Strike 1.8

User Manual

Schrödinger Press

June 2009

Contents

Document Conventions

In addition to the use of italics for names of documents, the font conventions that are used in this document are summarized in the table below.

Font	Example	Use
Sans serif	Project Table	Names of GUI features, such as panels, menus, menu items, buttons, and labels
Monospace	`$SCHRODINGER/maestro`	File names, directory names, commands, environment variables, and screen output
Italic	*filename*	Text that the user must replace with a value
Sans serif uppercase	CTRL+H	Keyboard keys

Links to other locations in the current document or to other PDF documents are colored like this: Document Conventions.

In descriptions of command syntax, the following UNIX conventions are used: braces `{ }` enclose a choice of required items, square brackets `[ ]` enclose optional items, and the bar symbol | separates items in a list from which one item must be chosen. Lines of command syntax that wrap should be interpreted as a single command.

File name, path, and environment variable syntax is generally given with the UNIX conventions. To obtain the Windows conventions, replace the forward slash / with the backslash \ in path or directory names, and replace the $ at the beginning of an environment variable with a % at each end. For example, `$SCHRODINGER/maestro` becomes `%SCHRODINGER%\maestro`.

In this document, to *type* text means to type the required text in the specified location, and to *enter* text means to type the required text, then press the ENTER key.

References to literature sources are given in square brackets, like this: [10].

Introduction to Strike

1.1 Strike Overview

Strike™ is a chemically-aware statistical package which is integrated with Maestro™ to provide a flexible and intuitive interface. Employing molecular data generated by Schrödinger software such as QikProp™, Glide™, Liaison™, or MacroModel®, or from other sources such as experimental data or third-party software, Strike can be used to do the following:

- Generate basic univariate and bivariate statistics such as mean, median, mode, covariance, and correlations
- Generate structure-activity relationship hypotheses using rigorous statistical methods
- Run validation tools to assess the validity and predictive power of generated QSAR/QSPR models
- Employ such models as filters and predictive tools
- Perform similarity analysis in molecular property or 2-dimensional structural space.

This document provides an introduction to Maestro, a set of tutorial exercises using the capabilities of Strike, a description of the Strike GUI in Maestro, a command line reference chapter, and definitions of some statistics terms and methods.

1.2 Running Schrödinger Software

To run any Schrödinger program on a UNIX platform, or start a Schrödinger job on a remote host from a UNIX platform, you must first set the `SCHRODINGER` environment variable to the installation directory for your Schrödinger software. To set this variable, enter the following command at a shell prompt:

csh/tcsh: `setenv SCHRODINGER` *installation-directory*

bash/ksh: `export SCHRODINGER=`*installation-directory*

Once you have set the `SCHRODINGER` environment variable, you can start Maestro with the following command:

```
$SCHRODINGER/maestro &
```

It is usually a good idea to change to the desired working directory before starting Maestro. This directory then becomes Maestro's working directory. For more information on starting Maestro, including starting Maestro on a Windows platform, see Section 2.1 of the *Maestro User Manual*.

1.3 Citing Strike in Publications

The use of this product should be acknowledged in publications as:

Strike, version 1.8, Schrödinger, LLC, New York, NY, 2009.

Strike Tutorial

This chapter is designed to help you become familiar with the functionality of Strike 1.8. Once you have worked through these exercises, you will have an understanding of the basic Strike features.

The Strike workflow for QSAR model generation/validation generally consists of three steps: data preparation, model generation and validation, and model application. The Strike workflow for similarity analysis using molecular properties also consists of three steps: data preparation, similarity calculation, and application of calculated similarities. For similarity analysis using two-dimensional structures (atom-pair similarity), two steps are required: the similarity calculation and application of calculated similarities. These steps will be illustrated in the tutorial exercises, which demonstrate how to do the following:

- Generate or import molecular data into Maestro for use by Strike
- Generate, validate, and apply QSAR/QSPR models
- Perform similarity analysis

Three tutorial examples are provided to demonstrate Strike workflows:

- Generating and testing a QSPR model for estimating aqueous solubility using a small number of molecular properties
- Developing a QSAR model for predicting activities of folate-based thymidylate synthase ligands
- Calculating similarities using 2-dimensional structures and molecular properties, and with these similarities extracting known actives for thermolysin from a ligand dataset.

To perform these exercises, you must have access to an installed version of Maestro 9.0 and Strike 1.8. For installation instructions, see the *Installation Guide*.

2.1 Creating a Working Directory and Copying Files

Before you begin the tutorial you need to create a working directory to keep all your input and output files, and then make a copy of the tutorial files.

UNIX:

1. Set the `SCHRODINGER` environment variable to the directory in which Maestro is installed:

csh/tcsh: `setenv SCHRODINGER` *installation_path*

sh/bash/ksh: `export SCHRODINGER=`*installation_path*

2. Change to a directory in which you have write permission.

   `cd` *mydir*

3. Create a directory by entering the command:

   `mkdir` *directory-name*

4. Copy the files to your working directory (*version* is the 5-digit Maestro version number):

   `cp -r $SCHRODINGER/maestro-v`*version*`/strike/tutorial/*` *directory-name*

Windows:

1. Open the folder in which you want to create the folder that serves as your working directory.

   The default working directory used by Maestro is your user profile, which is usually set to `C:\Documents and Settings\`*username*. To open this folder, do the following:

   a. Choose Run from the Start menu.

   b. Enter `%USERPROFILE%` in the Open text box and click OK.

2. Click Make a new folder under File and Folder Tasks.

   You can also choose Folder from the New submenu of the File menu.

3. Enter a name for the folder.

   If you want to create a folder inside this folder, repeat steps 1 through 3.

4. Open the folder that contains the tutorial files:

   a. Choose Run from the Start menu.

   b. Enter `%SCHRODINGER%` in the Open text box and click OK.

c. Open the `maestro-v`*version* folder (*version* is the 5-digit Maestro version number), then open the `strike` folder, then open the `tutorial` folder.

5. Select all the folders in the `tutorial` folder, and drag them to the folder you created in Step 3.

   You can close the `tutorial` folder after copying the files.

You now have working copies of the necessary files. In the following chapters, references to tutorial files in the `qsar`, `qspr`, and `simil` directories are to the files and directories in your working directory.

2.2 Starting Maestro and Setting the Working Directory

Once you have created the working directory you can start Maestro, and set the Maestro working directory. By default, Maestro writes job files to its working directory. You can change the default in the Preferences panel. If you have changed the default, you should change it back for this tutorial.

UNIX:

1. If you have not already done so, set the `SCHRODINGER` environment variable to the installation directory:

   csh/tcsh: `setenv SCHRODINGER` *installation_path*

   bash/ksh: `export SCHRODINGER=`*installation_path*

   This environment variable is also required to run Strike jobs.

2. Change to the desired working directory:

   `cd` *directory-name*

3. Enter the following command:

   ```
   $SCHRODINGER/maestro &
   ```

   The Maestro main window is displayed, and the working directory is Maestro's current working directory. If you are using an existing Maestro session, you can change the directory by choosing Change Directory from the Maestro menu, navigating, to the appropriate directory and clicking Choose.

Windows:

1. Double-click the Maestro icon on the desktop.

   You can also use the Start menu. Maestro is in the Schrödinger submenu.

2. Choose Change Directory from the Maestro menu.
3. Navigate to the appropriate directory and click Choose.

2.3 Generating and Testing a QSPR Model for Aqueous Solubility

The aqueous solubility of organic molecules plays a key role in ADME processes, especially absorption, distribution, and excretion. To experimentally measure accurate aqueous solubilities (logS) is difficult and requires a synthesis of the compound of interest. Because of this, a number of *in silico* approaches have been developed to estimate this key molecular property, including fragment-based approaches, linear models, and non-linear models. QikProp, Schrödinger's molecular property predictor which estimates 44 molecular properties, uses a linear method for estimating logS.

The QikProp model, as with all linear or non-linear models, was fit to a finite set of compounds. When examining molecules outside the chemical space used in the fitting process, high accuracy in logS predictions might not be obtained. Consequently it may be desirable to generate local QSPR (quantitative structure-property relationship) models relevant to the compounds of interest. This tutorial provides an example of generating a local model for logS prediction using only molecular properties.

Before you begin this exercise, change to the `qspr` directory:

1. Choose Change Directory from the Maestro menu.
2. Select `qspr` and click Choose.

2.3.1 Importing Data

1. Click the Import structures button on the toolbar.

   

2. In the Import panel, select the Maestro-format structure file `aq_sol_ligs.mae`.

   This file contains 1144 molecules for which experimental measurements of logS have been taken, as well as a set of calculated properties for each molecule.

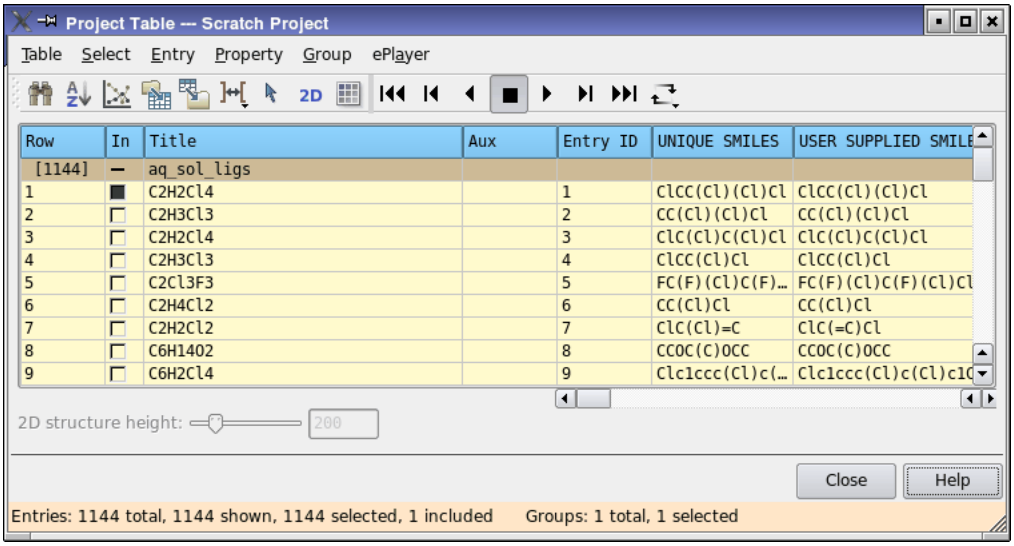

Figure 2.1. The Project Table after importing 1144 structures

3. Click Options.

   The Import Options dialog box opens.

4. Ensure that Import all structures is selected, and that the Include in Workspace option selected is First Imported Structure.

5. Click Close.

   The Import Options dialog box closes.

6. In the Import panel, click Open.

   The 1144 molecular structures in the file are imported. The import operation may take a minute to finish. When it has finished, the first structure in the file is displayed in the Workspace.

7. Click the Open/Close project table button on the toolbar.

   

   The Project Table panel opens.

As shown in Figure 2.1, each structure in the imported file is now an entry in the Project Table, represented by a row. The selected entries counter in the upper right corner of the panel reads 1144 selected. A long series of columns displays a number of molecular properties, or *descrip-*

tors, which were calculated in advance for each entry. All but the first four columns (Row, In, Title, and Aux) can be scrolled into or out of view. The last of these can be hidden by right-clicking in the column header and choosing Hide from the shortcut menu. The Project Table panel has been resized and these columns hidden for this and subsequent figures.

Strike does not generate descriptors. The descriptors in this exercise came from three sources:

- Most of the descriptors in the table were determined by QikProp, a program distributed by Schrödinger that generates a widely applicable set of molecular properties. For more information on QikProp, see the *QikProp User Manual*.
- A few descriptors were obtained from the `ligparse` utility (`$SCHRODINGER/utilities/ligparse`), including the Aromatic proportion and the Non-carbon proportion. The aromatic proportion is the fraction of heavy atoms that are aromatic while the non-carbon proportion is the fraction of heavy atoms that are not carbon.
- Also included are experimentally determined logS values in the measured log(solubility:mol/L) descriptor.

2.3.2 Preparing Test and Training Sets

The next step is to separate the 1144 molecules into two sets, a test set and a training set, using a random selection method that is part of the Project Table facility.

1. In the Project Table, choose Select > Random.

   The Random Selection dialog box opens.

2. Ensure that the value in the Randomly select *n* % of entries text box is 50, the default.

   By default, the random set is chosen from only the selected entries. When the structure file was imported, all entries in the project table were selected, but this may not always be the case.

3. Change the Select from option from Selected entries to All entries and click Select.

   After a moment, the Project Table is redisplayed with random entries selected. The selected entries counter in the bottom of the panel now reads 572 selected.

To keep track of the newly selected entries, which will be used as the training set, add a column to the Project Table that labels the currently selected molecules:

1. Choose Property > Add to open the Add Property panel.
2. In the Name text box, type `Population`.
3. Choose String from the Type option menu.

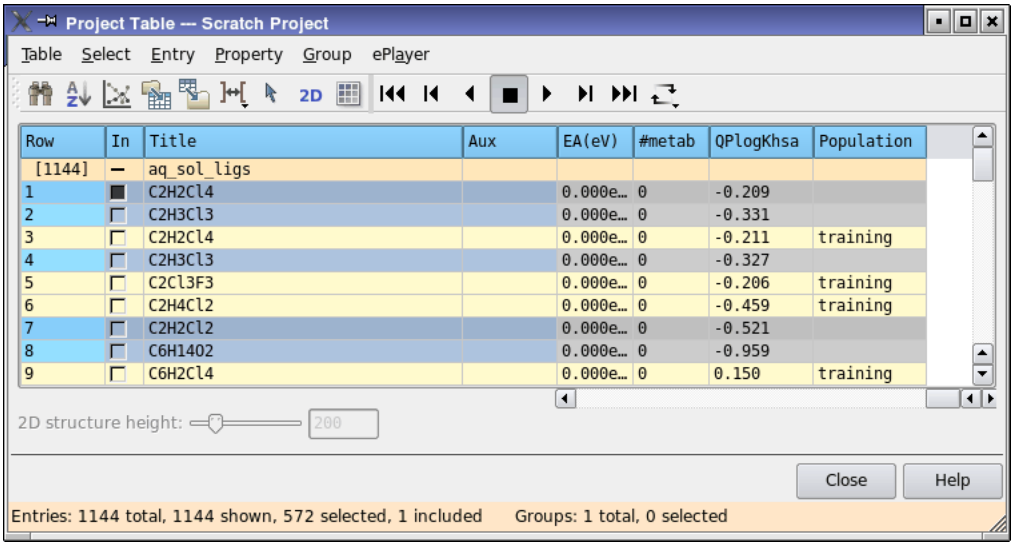

Figure 2.2. The Project Table with a randomly selected training set

4. In the Initial value text box, type `training`. Click Add.

   A column is added to the Project Table to the right of QPlogKhsa, as shown in Figure 2.2. Scroll to the far right to see this column. Under the column header Population, only the currently selected entries have a value of training.

Because the random selection generator is machine-dependent, your training set is unlikely to be a precise match to that shown in Figure 2.2, and therefore your results could differ from those shown in this document. Other results will also differ slightly because of differences in the random selections made.

The data has now been prepared. In the next section, it will be used to generate a model.

2.3.3 Building a Partial Least Squares Model

It is known from the general solubility equation that a relationship exists between a compound's aqueous solubility and its logP and melting points. We will use this idea in generating our model by including the logP estimate from QikProp along with a handful of molecular properties chosen to fulfill the role of the melting point.

Your first model will use the Partial Least Squares (PLS) method, which is described briefly in Chapter 5. Linear equations are generated that describe the relationship between a group of factors (derived from a set of independent descriptors) and a dependent descriptor (the predicted property). The goal of PLS is to find factors that explain the variance in both the independent and the dependent descriptors.

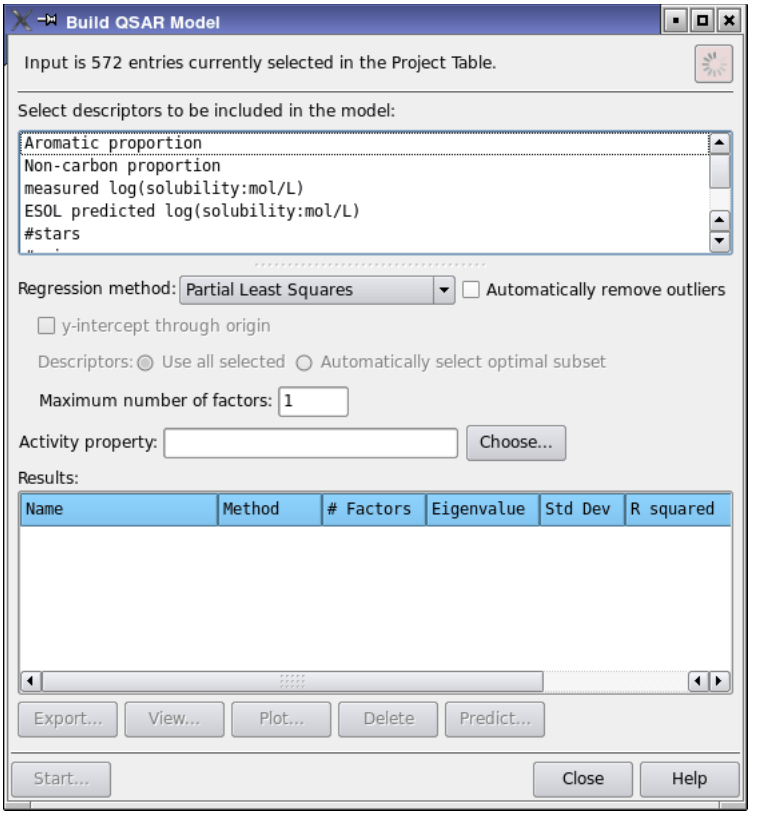

Figure 2.3. The Build QSAR Model panel

1. In the main Maestro window, choose Applications > Strike > Build QSAR Model.

   The Build QSAR Model panel opens. As shown in Figure 2.3, the input counter under the panel title bar reads Input is 572 entries currently selected in the Project Table.

2. Ensure that the Regression method selected is Partial Least Squares.

3. Under Select descriptors to be included in the model, control-click on the following:

   - Aromatic proportion
   - #rotor
   - volume
   - QPlogPo/w

   The descriptor count is displayed: (4 currently selected) These four descriptors will be the independent variables.

4. Ensure that Automatically remove outliers is deselected (the default).

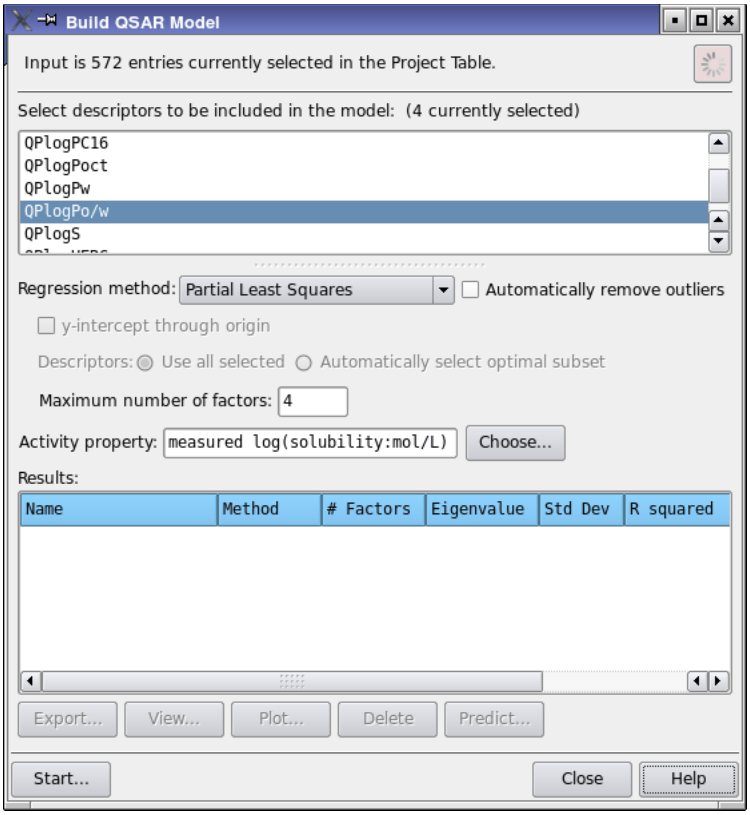

Figure 2.4. The Build QSAR Model panel settings for the PLS model

5. Enter 4 as the Maximum number of factors.

   The Maximum number of factors should be less than or equal to the number of independent variables. If the Maximum number of factors is greater than the number of independent variables, Strike will automatically report the maximum number of factors extracted from the data, which is generally equal to the number of independent descriptors.

6. Select the Activity property (the dependent variable to be fit) by clicking Choose.

   The Choose Activity Property dialog box opens.

7. Select measured log(solubility:mol/L) and click OK.

   These settings mean that the model will attempt to correlate the number of rotatable bonds (#rotor), fractional aromatic proportion, molecular volume and logP (QPlogPo/w) to experimentally measured aqueous solubilities (measured log(solubility:mol/L)).

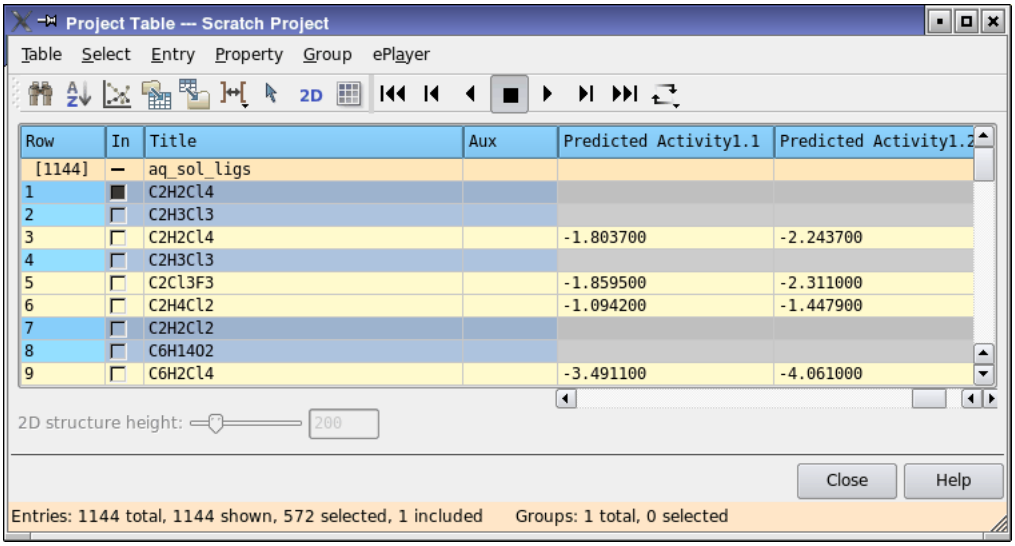

Figure 2.5. The Project Table with training-set predicted activities

8. Click Start.

   A Start dialog box opens.

9. Name the job `solubility`.

10. Select the Job options you want, then click Start to begin the calculation.

    The Monitor panel opens as the job begins to run.

    The job takes only a few moments to finish. When the model has been generated, the results are incorporated into the Project Table, shown in Figure 2.5.

2.3.4 Examining PLS Model-Building Results

In the Project Table there are four new columns, headed Predicted Activity*X.Y*, where *X* represents the model and *Y* the number of factors used in the prediction. This is the first model built in this project, so $X = 1$, and a maximum of 4 factors were used, so $Y = 1, 2, 3,$ or 4. The values in each column are the predicted values of logS generated using *Y* factors.

In this document, the particular set of diagnostic statistics generated by model *X* with *Y* factors is called a *predictor*. Four predictors are listed in the Results table of the Build QSAR Model panel after the model-building job has finished, as shown in Figure 2.6. Along with the Name in the format *jobname.X.Y*, the Method, and the number of factors (# Factors), statistical information is given for an immediate appraisal of the predictors: standard deviation, R-squared, F-value, and P-factor.

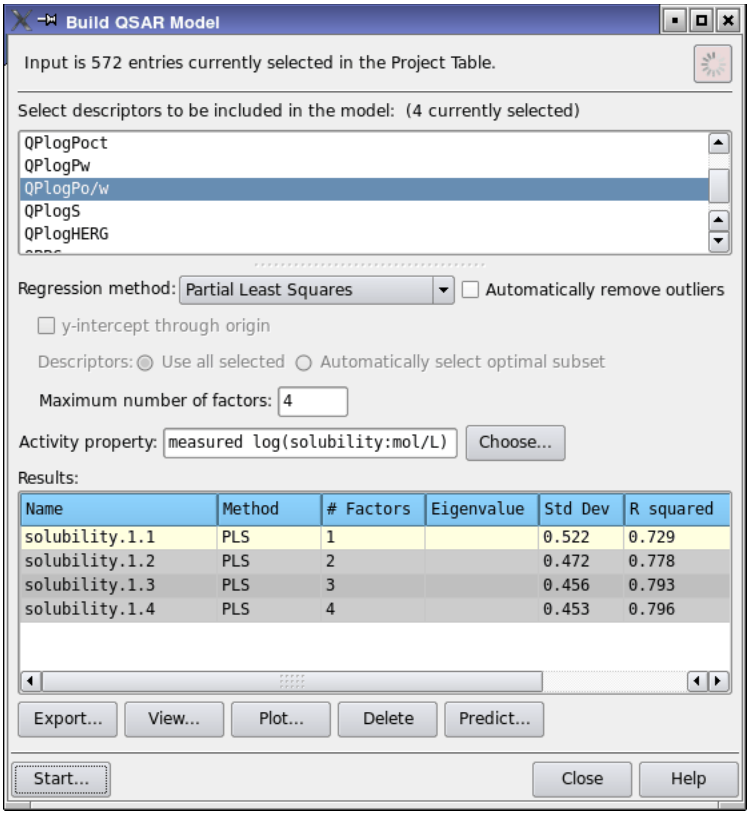

Figure 2.6. The Build QSAR Model panel with a four-predictor PLS model

The five buttons below the Results table are now available. When a predictor is selected, the buttons can be used to perform the following tasks. Some tasks affect only the selected predictor; others operate on the model as a whole, including any other predictors belonging to the model:

Export	Export the model for use in another project
View	View the output file for the model-building job that generated the predictor
Plot	Plot the predicted versus experimental results for the selected predictor
Delete	Delete the model that generated the predictor
Predict	Make further predictions using the selected predictor

More information about the model and the predictors is given in the output file of the model-building job, *jobname*`.out`, which can be examined using the View button:

1. In the Build QSAR Model panel, click the View button.

   The View QSAR Model dialog box opens. This dialog box displays the output file for the Strike model-building job—see Figure 2.7.

2. Examine the output file, noting the following points of interest:

- The Correlation Matrix for input variables (the four independent descriptors).
- PLS Regression Statistics, listing standard deviation (S.D.), R-squared, F-factor, and P-value by #Factors. The large F-factors and small P-values indicate this model was likely not achieved by chance and that the descriptors chosen are significant as a set.
- Cross Validation leave-*N*-out Results over *M* Cycles. Large differences between calculated q^2 and r^2 values reflect significant dependence of the model on the molecules included in the regression and in general are unfavorable.
- T-values and coefficients.
- Predicted values for logS at each #Factors for each of the 572 molecules in the training set.

3. Click OK to close the View QSAR Model dialog box.

If a job fails, the View button will not display the output file. Examine the output file *jobname*`.out` in a text editor or the Monitor panel instead.

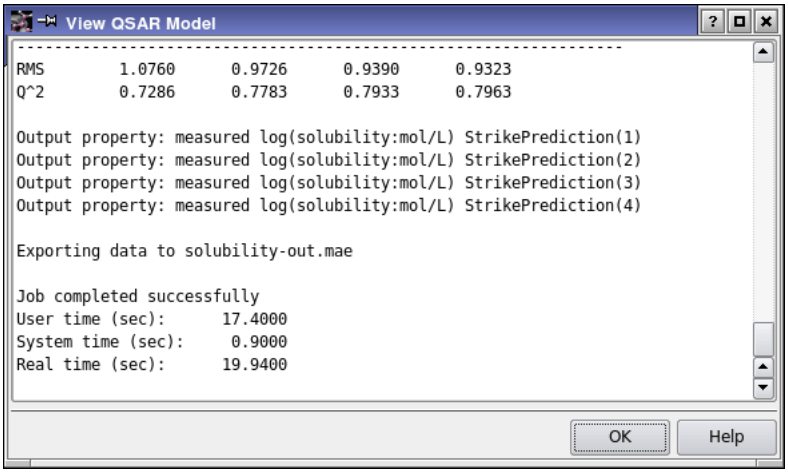

Figure 2.7. The View QSAR Model dialog box

2.3.5 Applying the Model to the Test Set

The true test of any model is to check its predictions against a set of molecules not included during its training. The exercise performed in this section would typically be considered part of model generation and validation, but for the purposes of this tutorial, it will be used to demonstrate the model application step of the Strike workflow.

The first step is to create a test set of molecules. In this example, the test set will be those molecules in the Project Table that were not members of the training set:

1. In the Project Table, confirm that the training set is selected by examining the Population column. If so, skip to the next step.

   If for any reason the training set is no longer the selected set—for example, if a single entry has been selected instead—you can restore the selection by performing these steps:

   a. Choose Select > Only from the Project Table.

      The Entry Selection panel opens.

   b. In the Properties list, select Population.

   c. Select the option Is defined (any value).

      Only the training set has a defined value (training) in the Population column.

   d. Click Add, then OK.

   The molecules in the training set, and only those molecules, are now selected.

2. In the Project Table, choose Select > Invert.

   The selected molecules are now those that were not part of the training set. This will be the test set.

Run a prediction job on the test set molecules:

1. In the main window, choose Applications > Strike > Predict.

   The Predict based on QSAR model panel opens. If it was open, the Build QSAR Model panel closes.

   In the Predict panel, the four predictors previously generated are listed in the Select model to use for prediction table.

2. Select the model with 4 in the # Factors column.

3. Click the Start button to open the Start dialog box. Change the Host and Username if necessary before clicking Start to launch the `strike_predict` job.

   The Monitor panel appears. The job takes a few seconds to run.

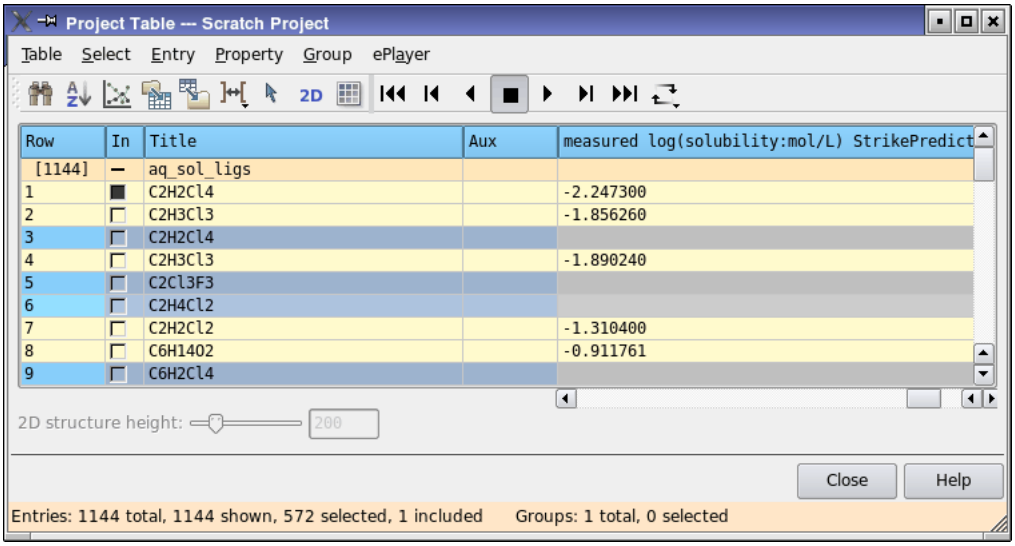

Figure 2.8. The Project Table with test-set predicted activities.

4. When the job is finished, view the Project Table.

   There are four new columns to the right of the table: measured log(solubility:mol/L) StrikePrediction(*N*), where *N*=1, 2, 3, or 4. These columns hold predicted logS values for the test set, as shown in Figure 2.8.

2.3.6 Calculating Univariate and Bivariate Statistics

The Strike statistics script calculates univariate and bivariate statistics of selected descriptors for the set of entries currently selected in the Project Table. The Strike Univariate and Bivariate Statistics panel allows you to select from a list of the descriptors found in the Project Table. When you have run a statistics job, the results are displayed in the dialog box, from which information can be copied and pasted to an open file as reference material or for printing.

If for any reason the test set is no longer the selected set—for example, if a single entry has been selected instead—you can restore the selection by performing these steps:

a. Choose Select > Only from the Project Table.

   The Entry Selection panel opens.

b. In the Properties list, select Population.

c. Select the option Is defined (any value).

   Only the training set has a defined value (training) in the Population column.

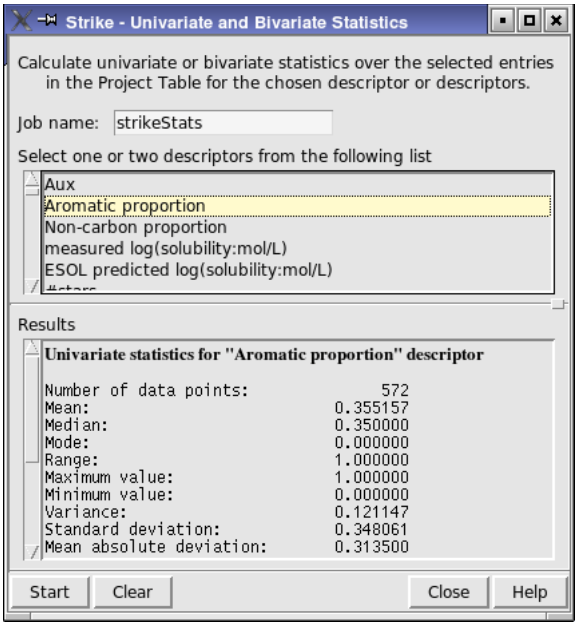

Figure 2.9. The Univariate and BIvariate Statistics panel with univariate statistics

d. Click Add.

e. Click Invert and then OK.

The molecules in the test set, and only those molecules, are now selected.

1. From the main Maestro menu, choose Applications > Strike > Statistics.
2. Select Aromatic_proportion from the list under Select one or two descriptors from the following list.

   The selected descriptor is highlighted in yellow. This will be a univariate statistics calculation, so the only input needed is the single descriptor you have selected.
3. Click Start to launch the job under the default name, strikeStats.

   After a moment, the job results appear in the Results text area, as shown in Figure 2.9. These univariate statistics describe the range and variance of the descriptor values and the shape of the distribution for the test set of molecules (the currently selected entries in the Project Table). See Chapter 5 for definitions of statistics terms.

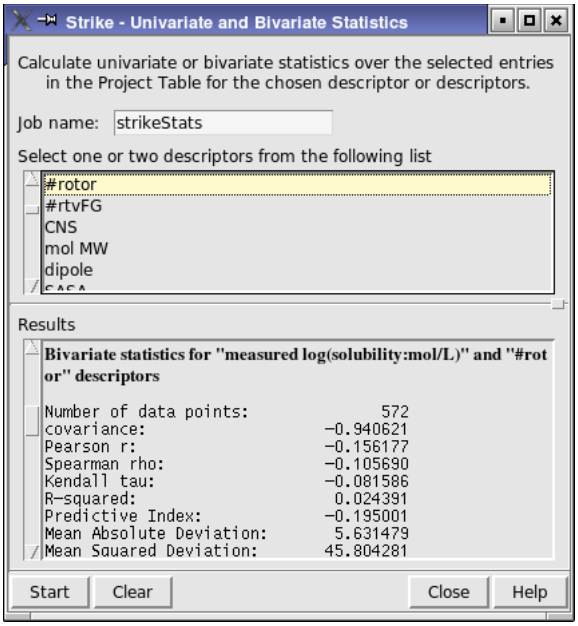

Figure 2.10. The Univariate and BIvariate Statistics panel with bivariate statistics

Now set up a bivariate statistics calculation.

4. From the descriptor list, select measured_log(solubility:mol/L), then control-click on #rotor.
5. Click Start.

   After a few moments, the new results are appended to the Results table. They include a small set of bivariate statistics for the pair of descriptors, followed by the univariate statistics for each, as shown in Figure 2.10. See Chapter 5 for definitions of statistics terms.

The dialog box obtains its descriptor information from the Project Table. If you subsequently add or remove properties (descriptors) from the Project Table, you need to close the statistics dialog box and then reopen it to capture the new information.

2.3.7 Model-Building Using Principal Component Analysis

In this exercise, you will generate a model using one of the other alternative regression methods available in Strike, Principal Component Analysis.

1. In the Project Table, the test set is currently selected. Choose Invert from the Select menu to select the training set.

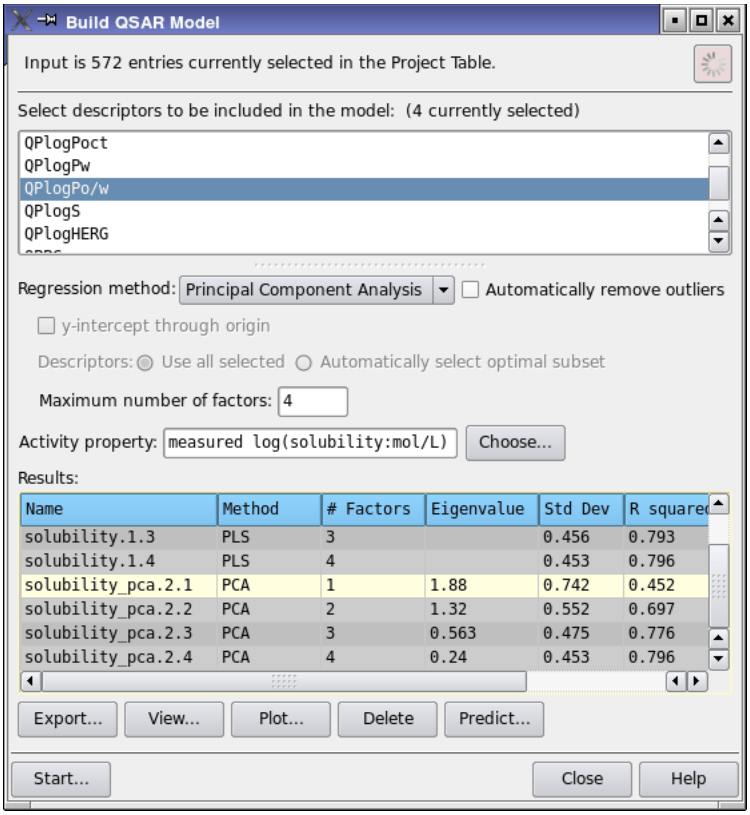

Figure 2.11. Build QSAR Model panel with four-predictor PCA model

2. Open the Build QSAR Model panel.

   The Predict panel closes. The Build QSAR Model panel retains the selected descriptors, number of factors, and regression method used to generate the previous model. The four predictors generated by the PLS model-building job remain in the Results table.

3. Choose Principal Component Analysis from the Regression method option menu and click Start.

4. In the Start dialog box, change the job name to `solubility_pca` and click Start to launch the calculation.

   When the job has finished, the new model will be added to the Results table as a set of predictors with # Factors equal to 1, 2, 3, and 4, as was the PLS model. In the Eigenvalue column, a number is associated with each of the four PCA-model predictors. The eigenvalue represents the portion of the total variance accounted for by the *n*-factor predictor.

In the Project Table, the four new columns Predicted Activity2.*n* are added to the table, with values only for the training set of molecules.

If you were using this PCA model in a real project, you could continue by carrying out the analysis and prediction steps that were performed for the PLS model earlier in this chapter.

The PCA method is frequently used for data reduction by retaining only those factors needed to account for most of the total variance. The variance of each of the independent variables used in the model is taken to be 1.0, and the total variance is defined as the sum of the variances of each independent variable. Typically it is sufficient to retain only those factors with an eigenvalue greater than 1.0. These are the factors that account for more of the variance than does any single original variable.

In the Results table in the Build QSAR Model panel, the *n*-factor predictor of a PCA model accounts for a portion of the total variance equal to the sum of the first *n* eigenvalues. For example, in the table shown in Figure 2.11, the first eigenvalue is 1.88 and the second 1.32, while the third and fourth eigenvalues are less than 1.0. The total variance is 4.0, and the two-factor predictor is sufficient to account for (1.84 + 1.32)/4.00 = 79% of the total variance.

2.3.8 Model-Building Using Multiple Linear Regression

The third regression method available for model-building in Strike is multiple linear regression (MLR). In this section, you will generate an MLR model that uses an algorithm to select the optimal set of descriptors for use.

1. Ensure that the training set is selected in the Project Table.
2. In the Build QSAR Model panel, use control-click to add two more descriptors, mol MW and SASA to the Select descriptors to be included in the model list.
3. Choose Multiple Linear Regression from the Regression method option menu.
4. Select the Descriptors option Automatically select optimal subset.
5. Ensure that the Size of optimal subset is 4.

   These settings instruct the MLR algorithm to use the best subset of four descriptors from the six selected.
6. Click Start.

   The Statistics / Build QSAR - Start dialog box opens.
7. Change the job name to `solubility_mlr`.
8. Select the other job options you want, and click Start to launch the calculation.

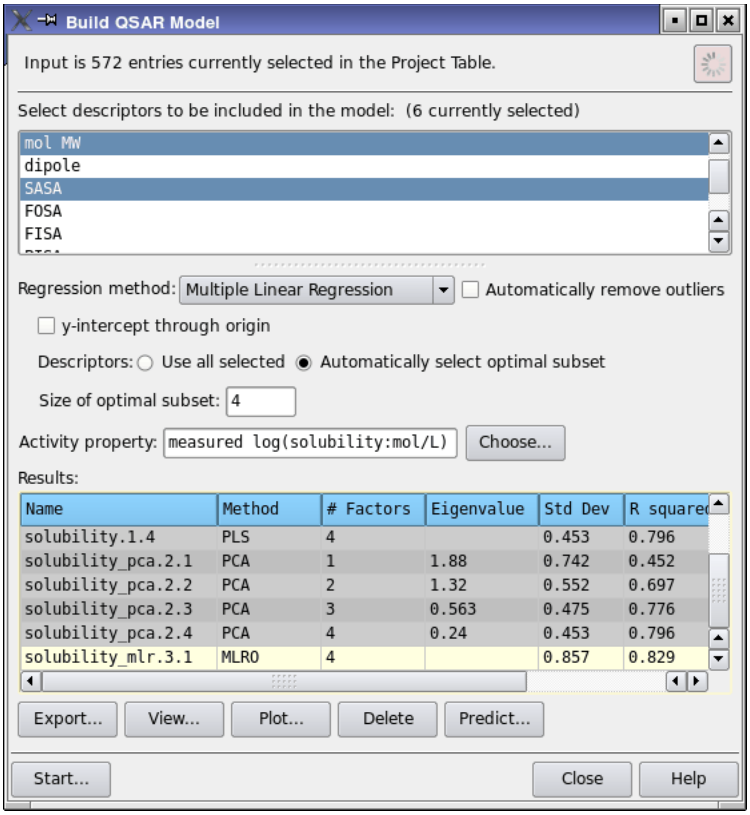

Figure 2.12. Build QSAR Model panel with MLR model

The job takes only a few moments to finish. When the model has been generated, it appears in the Results table in the Build QSAR Model panel as a single row with MLRO (MLR optimal subset) as the Method. See Figure 2.12.

The results are also incorporated into the Project Table as the column Predicted Activity3.1.

Again, you could continue with analysis and prediction using this model, as you did with the PLS model.

2.4 Calculating Atom-Pair Similarities

It is often useful to identify molecules that are "similar" in a chemically significant way to structures of interest. Strike can be used to analyze similarity in either two-dimensional structural (atom-pair connectivity) or molecular descriptor space. Using the atom-pair connectivity

method has the advantage of requiring only structural (connectivity) information for a set of probe molecules and for the molecules for which calculated similarities are desired.

In this section, you will use Strike to calculate atom-pair similarities, then use those similarities to extract known actives for thermolysin from a ligand data set.

2.4.1 Preparation

1. If Maestro is already running, choose Close from the Project menu.

   If the project is a scratch project from the previous exercises, you may discard it.

2. Choose Change Directory from the Maestro menu.
3. Navigate to your working directory.
4. Choose `simil` and click OK.

2.4.2 Importing Active and Decoy Ligands

First we need to create a ligand database which is seeded with a subset of the known active ligands. Using the remaining active ligands as probe molecules, we will attempt to extract the seeded actives out of the database.

1. Click the Import structures button in the toolbar.

   

2. In the Import panel, select the Maestro-format structure file `1tmn_actives.mae`.

   This file contains nine known active ligands for thermolysin.

3. Click Options.

   The Import Options dialog box opens.

4. Ensure that Import all structures is selected, and that the Include in Workspace option selected is First Imported Structure.
5. Click Close.

   The Import Options dialog box closes.

6. In the Import panel, click Open.

   The first active ligand structure appears in the Workspace.

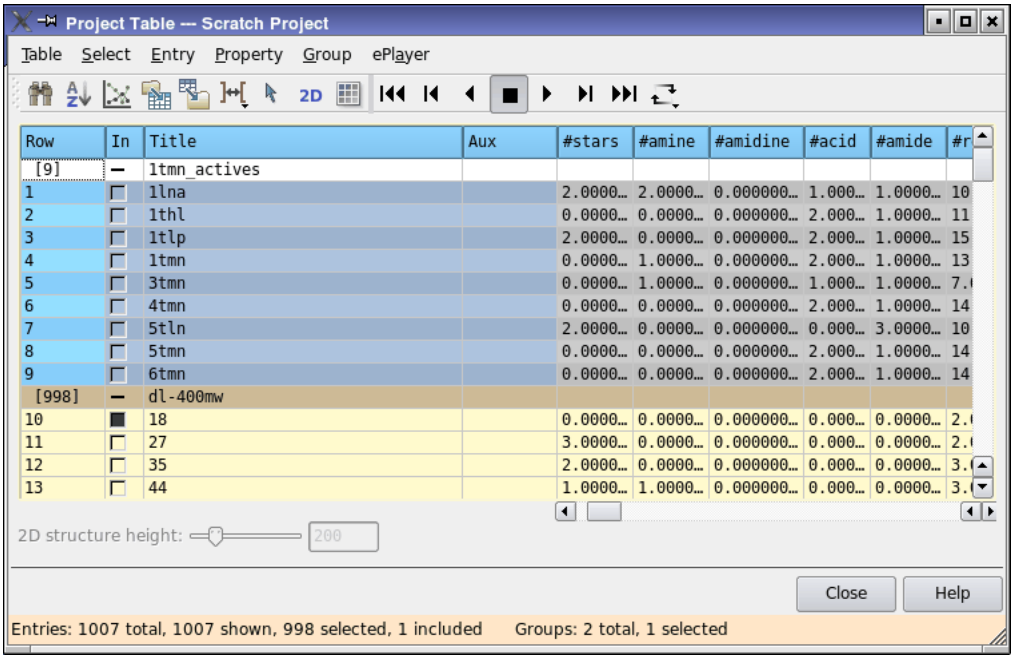

Figure 2.13. The Project Table with 9 active and 998 decoy ligands.

7. Open the Project Table panel.

   There are nine entries, each with a Title identifying the ligand. Each row also has columns of calculated molecular properties from QikProp. These properties will not be used in this exercise, as they are not needed to generate atom-pair similarities.

8. Display each of the ligands in turn in the Workspace.

   These structures show some diversity though many have peptide moieties.

   Note: if the Workspace appears empty, click the Fit to screen toolbar button

   

   to bring the ligand into view.

9. Click the Import structures button in the toolbar.

   

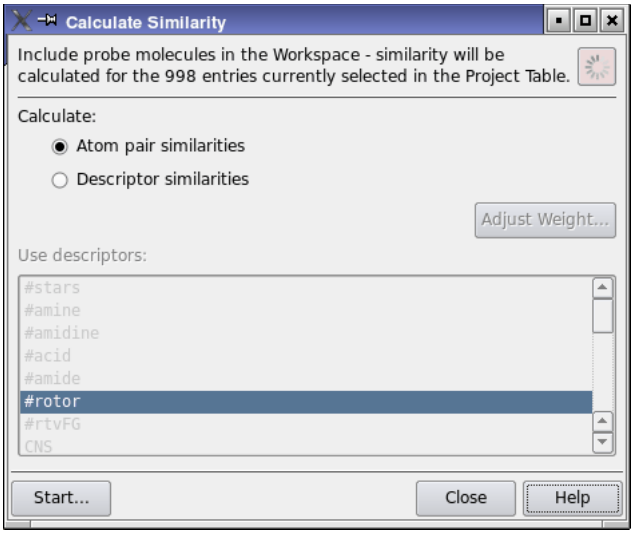

Figure 2.14. The Calculate Similarity panel

10. In the Import panel, select the file `dl-400mw.mae` and click Open.

    This file contains the Maestro-format structures of 998 decoy ligands with an average molecular weight of 400.

    After a few moments, the first decoy structure appears in the Workspace and 998 new entries are added to the Project Table. As can be seen in Figure 2.13, these entries also have molecular properties calculated by QikProp, which will not be used in this exercise.

2.4.3 Opening the Calculate Similarity Panel

1. Choose Applications > Strike > Similarity in the main window.

   The Calculate similarity panel opens, as shown in Figure 2.14. As noted in the panel, similarity will be calculated for the entries selected in the Project Table, using the entries included in the Workspace as probe molecules.

2. Ensure that the Atom pair similarities option is selected.

2.4.4 Seeding the Database and Designating Probes

At this point, all of the decoy ligands and none of the actives are selected in the Project Table. In this exercise, you will include three active ligands in the Workspace and add the other six active ligands to the selection to create the seeded ligand set.

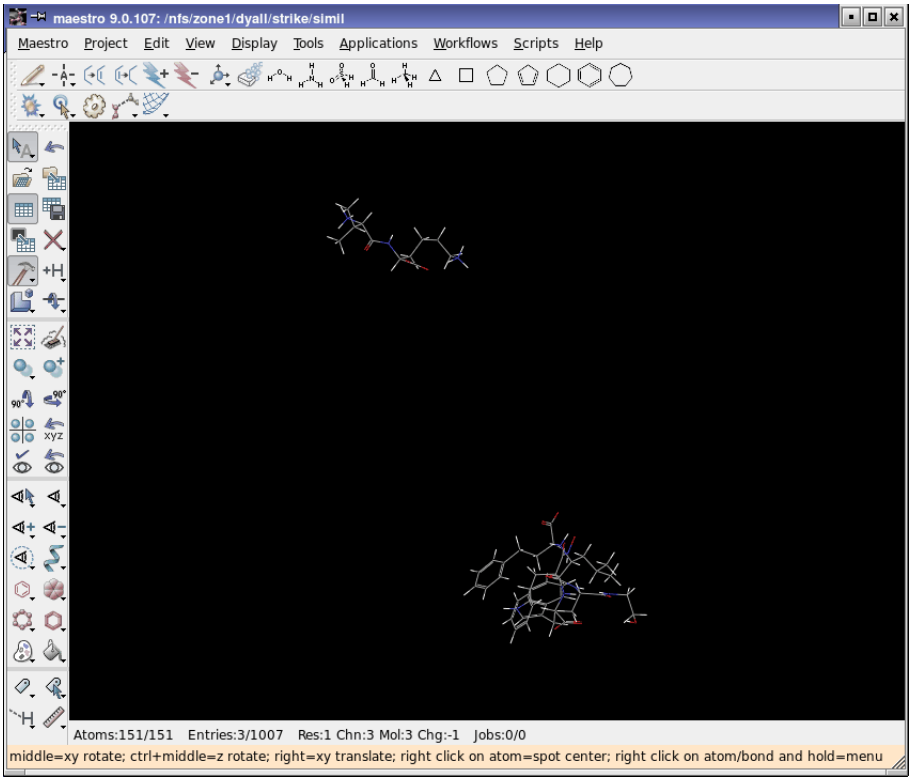

Figure 2.15. The three probe molecules included in the Workspace

1. Include the three active ligands, `1lna`, `1tmn`, and `5tln` in the Workspace

   Use control-click for the second and third ligands.

   These three entries are *not* selected in the Project Table.

   Figure 2.15 shows the three probe molecules. You may have to click the Fit to screen toolbar button to bring them into view.

2. Add the six active ligands that are not included in the Workspace (`1thl`, `1tlp`, `3tmn`, `4tmn`, `5tmn`, and `6tmn`) to the selected entries by control-clicking their rows.

   The database for which similarities will be calculated now contains 1004 entries, of which six are known actives. The resulting Project Table is shown in Figure 2.16.

3. In the Calculate Similarity panel, click Start.

   The Statistics / Similarity - Start dialog box opens.

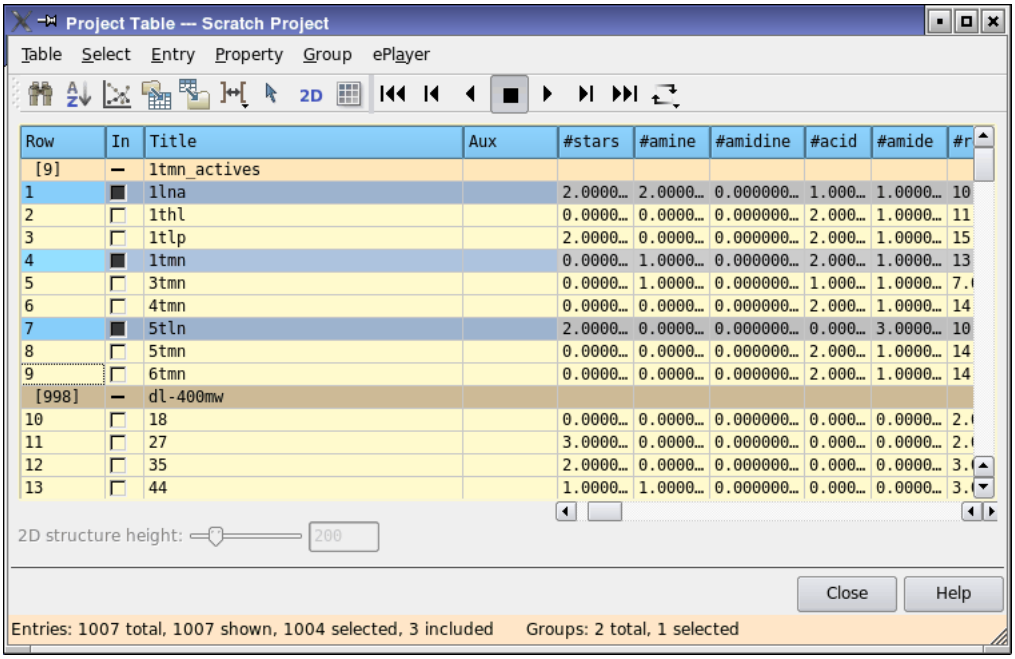

Figure 2.16. The Project Table with 1004 entries selected, 3 probes included.

4. Choose job options, then click Start to run the calculation.

   The Monitor panel appears as the job begins to run.

Two columns are added to the Project Table upon completion of the job. For each entry, the Max AP Similarity is the maximum atom-pair similarity of that structure to any of the probe molecules, and the Mean AP Similarity is the mean of the atom-pair similarities to each of the probes.

By default, atom-pair similarities are calculated on a scale from 0.0 to 1.0, with 0.0 indicating no structural similarity and 1.0 indicating maximum structural similarity.

2.4.5 Applying Atom-Pair Similarity

In this exercise, you will examine how well atom-pair similarity performs in extracting the six active ligands (those not used as probes) from the set of decoy ligands. To do this, you will sort the entries in the Project Table by similarity. Project Table entries (rows) can be sorted by multiple user-specified properties (columns) called primary, secondary, and tertiary keys.

When you imported the entries, they were grouped according to the file they were imported from. To sort the entries they must first be ungrouped.

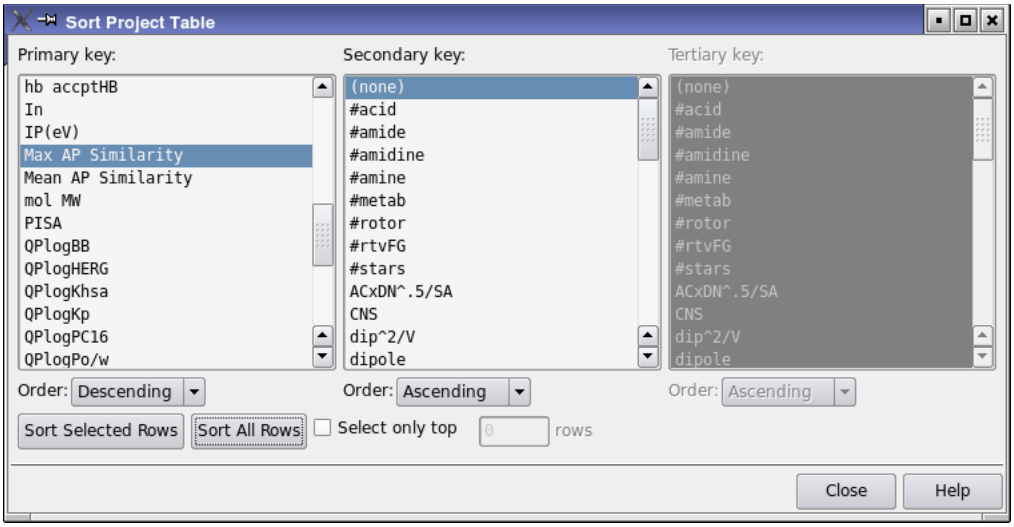

Figure 2.17. The Sort Project Table panel.

1. In the Project Table, select the 1tmn_actives group, and choose Entry > Ungroup.
2. Click Yes when prompted to delete the empty group.
3. Select the dl-400mw group, and choose Ungroup from the Entry menu.
4. Click Yes when prompted to delete the empty group.
5. Click the Sort button on the Project Table panel toolbar.

   

   The Sort Project Table panel opens, as shown in Figure 2.17.
6. Choose Max AP Similarity from the Primary Key list.
7. Choose Descending from the Order option menu under the Primary Key list.
8. Click Sort All Rows.

   The Project Table is sorted by descending Max AP Similarity, as in Figure 2.18.

   All six actives were found within the first 41 compounds (4.1% of the data set), and five in the first 28 compounds (2.8% of the data set). The probe molecules are at the end of the Project Table.

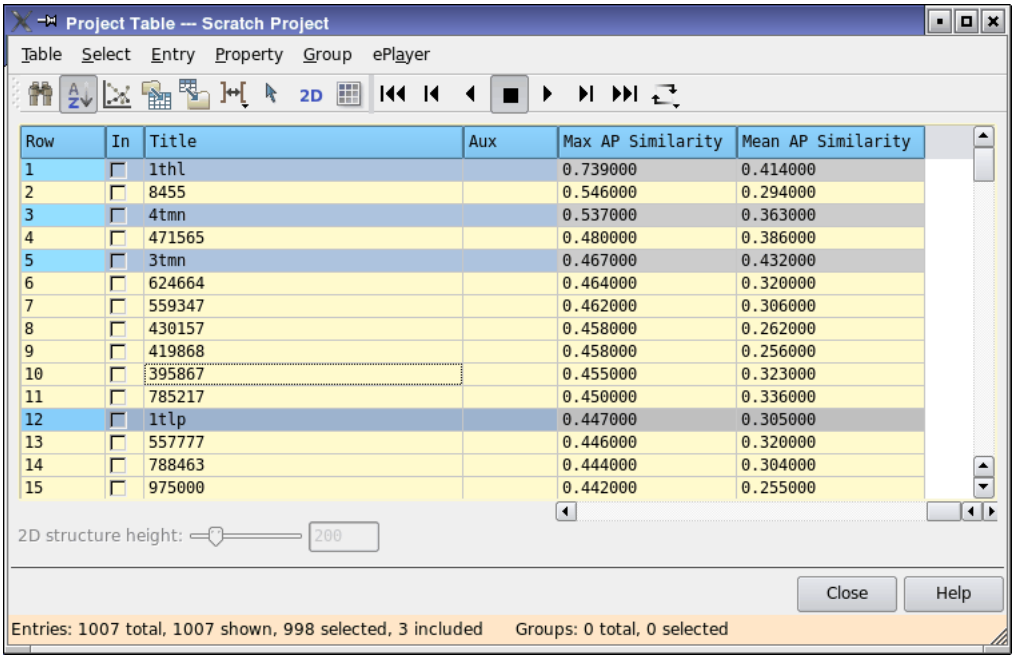

Figure 2.18. The Project Table sorted by Max AP Similarity.

9. As a second test, sort the Project Table again, repeating the previous steps, but this time using Mean AP Similarity as the Primary Key.

   Using the mean atom-pair similarities, all six actives are found in the first 21 compounds (2.1% of the data set).

It is not surprising that the mean atom-pair similarity does a better job of extracting actives from the data set than the maximum atom-pair similarity. Because all actives are at least slightly structurally similar, their mean values are raised compared to decoy ligands, which may share common features with only one active molecule.

This example shows that Strike can be used to extract compounds similar to a set of molecules using 2D-geometry atom-pair similarities. Next, you will perform this extraction using descriptor similarity instead of atom-pair similarity.

2.5 Calculating Descriptor Similarities from Molecular Properties

Now you will use calculated molecular properties to test the ability to extract actives from the data set using descriptor similarities. The molecular properties for the thermolysin active ligands and decoy ligands were previously determined using QikProp. From this set of molecular properties, four descriptor-based similarities can be calculated: *Euclidean similarity*, *Euclidean squared similarity*, *Manhattan similarity*, and *Tanimoto similarity*. Each of these methods calculates the descriptor-space distance between two molecules as a function of their molecular properties. For a summary of each of these methods, see Chapter 5.

The calculated similarities for all but Tanimoto similarity are expressed as distances on an arbitrary scale, where the smaller the value (the shorter the distance), the more similar the two molecules. High values for these quantities correspond to longer distances in descriptor space, indicating less similarity.

The Tanimoto similarity is calculated on a scale from 0.0 to 1.0 with 1.0 indicating maximum similarity and 0.0 indicating no similarity.

Like atom-pair similarity, Strike calculates descriptor similarities for a set of molecules relative to one or more probe molecules. The molecules included in the Workspace are used as probes. In this example, the probes will also be included in the test set.

1. In the Project Table, choose Select > All to select all entries.
2. Ensure that ligands `1lna`, `1tmn`, and `5tln` are included in the Workspace.
3. If the Calculate similarity panel is not open, open it by selecting Applications > Strike > Similarity from the main window.
4. Select Descriptor similarities.

   The Use descriptors list becomes available for choosing molecular properties to use in calculating similarities.
5. Select the donor HB and hb accptHB descriptors as shown in Figure 2.19 and click Start.

   The Start dialog box opens.
6. Change the job options if necessary, then click Start to run the calculation.

   After a few seconds, the Monitor panel reports that the job has finished. In the Project Table, the calculated descriptor similarities have been added as properties. These properties include maximum and minimum distances as well as the similarity measures.

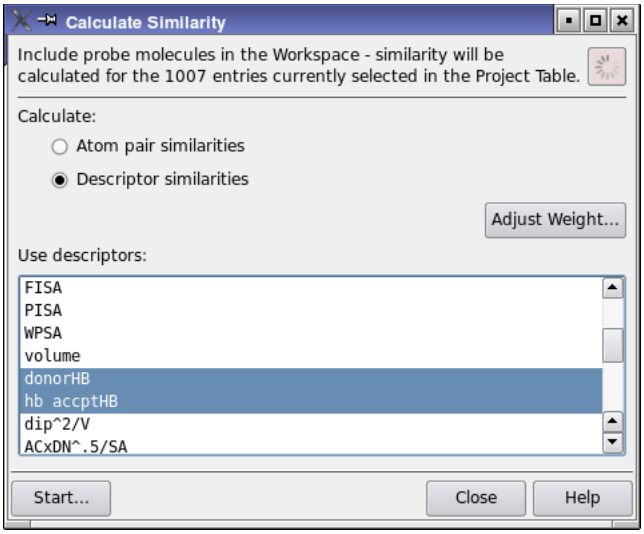

Figure 2.19. The Calculate Similarity panel with descriptor similarities specified

In the descriptor similarity calculation, the molecular properties of the probe molecules are averaged so they can be treated as a single virtual probe molecule. Similarity is calculated with respect to only the selected properties. You will now examine how well descriptor similarity based only on the number of hydrogen-bond acceptors and donors extracts known actives from the data set.

7. If the Sort Project Table panel is not open, click the Sort button on the Project Table panel toolbar.

   

   The Sort Project Table panel opens.

8. Select Euclidean sq, set the Order to Ascending, and click Sort All Rows.

   The smallest values, corresponding to the greatest similarity to the probes, appear at the top of the table. Of the 6 non-probe active ligands, 5 are found within the top 375 ligands. The remaining active, 1tlp, is ranked as last, due to its very large number of hydrogen-bond acceptor sites.

9. Now sort based on Tanimoto in Descending order.

   Of the 6 non-probe actives, 5 are found in the top 275 compounds. The 1tlp ligand is again the lowest-ranked active.

10. Close the Sort Project Table and the Calculate Similarity panels.

These very simple examples were designed to show possible applications of Strike similarity calculations in descriptor and 2D-similarity space.

2.6 Estimating Activity by Creating a QSAR Model

In this tutorial, you will build a QSAR model and use it to predict activity. The most significant difference between this exercise and the QSPR model-building exercise in Section 2.3 on page 6 is that the property being predicted is a biological activity. The workflow and steps that follow are similar to the QSPR exercise.

Thymidylate synthase is an anticancer drug target as it catalyses the generation of deoxythymidine monophosphate given dUMP and a cofactor, 5,10-methylene tetrahydrofolate, an essential step in de novo DNA replication. The widely used anticancer agent 5-fluorouracil targets thymidylate synthase and is active against solid tumors like breast, head, neck, and colon cancers. Activities (EC_{50} and IC_{50}) have been experimentally determined for a large series of compounds. You will use the set of broadly applicable descriptors generated by QikProp in order to develop a QSAR model with Strike.

For this tutorial, a set of 188 known inhibitors were selected. All of these ligands have experimentally-determined L1210 IC_{50} activities that range from 141 to 0.00052 µM. This set of ligands is well suited for QSAR as the structures have similar cores with a large variety of substitution in shared sidechains which leads to a wide activity range. The ligands were prepared from 2-D geometries using LigPrep, then neutralized, prior to being run through QikProp to produce 36 predicted properties.

The Maestro format file `thymidylate_synthase_ligands.mae` contains the 188 ligand structures with their QikProp properties, their raw IC_{50} values, and their -log(IC_{50}) values. Because the goal is a free energy relationship between activities and properties, you will use the -log(IC_{50}) or log(1/IC_{50}) values rather than the raw IC_{50} values.

2.6.1 Preliminaries

1. If Maestro is already running, choose Close from the Project menu.

   If the project is a scratch project from the previous exercises, you may discard it.

2. Choose Change Directory from the Maestro menu.
3. Navigate to your working directory.
4. Choose `qsar` and click OK.

2.6.2 Preparing the Data

The ligand data must be imported into the project and divided into a test set and a training set.

1. Click the Import structures button on the toolbar.

2. In the Import panel, navigate to the `qsar` directory and select the file `thymidylate_synthase_ligands.mae`.
3. Click Options.

   The Import Options dialog box opens.
4. Ensure that Import all structures is selected, and that the Include in Workspace option selected is First Imported Structure.
5. Click Close.

   The Import Options dialog box closes.
6. In the Import panel, click Open.

   After a moment, the first ligand in the file appears in the Workspace.
7. Open the Project Table panel.

   There are 188 entries in the project, all selected.
8. In the Project Table choose Select > Random.

   The Random Selection dialog box opens.
9. Ensure that the value in the Randomly select *n* % of entries text box is 50, the default, and click Select.

   After a moment, the Project Table is redisplayed with half the entries deselected at random. The selected entries counter in the upper right corner of the panel now reads 94 selected.

To keep track of the newly selected entries, which will be used as the training set, add a column to the Project Table that labels the currently selected molecules:

1. Choose Property > Add to open the Add Property panel.
2. In the Name text box, type `Population`.
3. Choose String from the Type option menu.

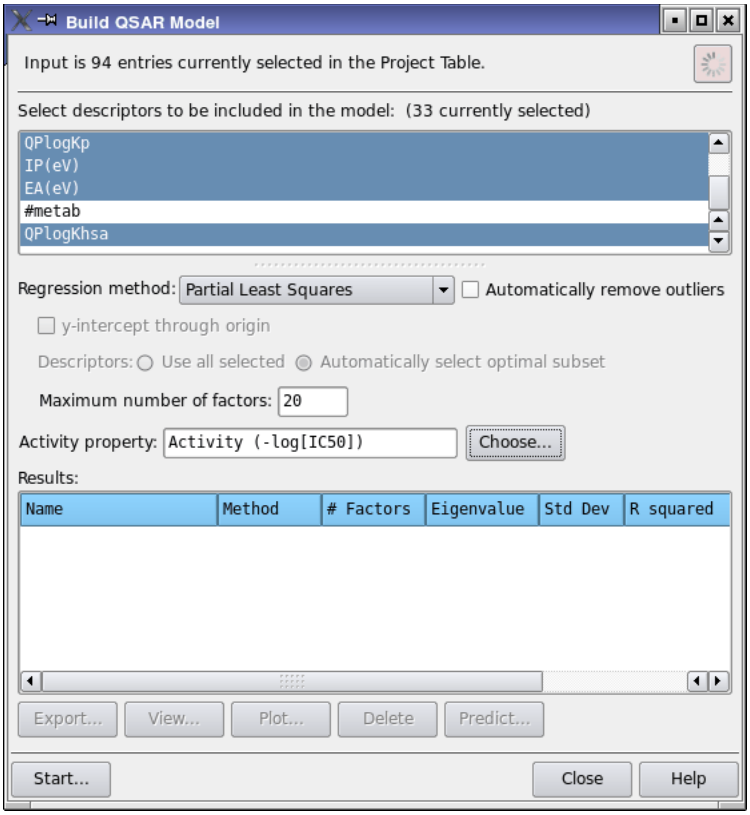

Figure 2.20. Build QSAR Model panel showing settings for activity model

4. In the Initial value text box, type `training`.
5. Click Add.

   Under the column header Population, only the currently selected entries have a value of training.

2.6.3 Model Generation

You will now build a QSAR model employing all of the relevant QikProp descriptors:

1. Choose Applications > Strike > Build QSAR Model in the main window.

   The Build QSAR Model panel opens. The input counter under the title bar reads Input is 94 entries currently selected in the Project Table.

2. Under Select descriptors to be included in the model, select all the descriptors.

3. Control-click to deselect the following descriptors: Activity (-log[IC50]), #stars, #rtvFG, CNS, QPlogBB, and #metab.

   The latter five descriptors are omitted because they are expected to be unrelated to the binding process; the first will be the dependent variable.

4. Ensure that the Regression method is Partial Least Squares and that Automatically remove outliers is not selected.

5. In the Maximum number of factors box, type `20`.

6. Click Choose.

   The Choose Activity Property dialog box opens.

7. Select Activity (-log[IC50]) from the list and click OK.

   See Figure 2.20 to check your settings.

8. Click Start.

   The Start panel opens.

9. Check that the job name is `strike_buildqsar`.

10. Change the job options if necessary, then click Start to begin the job.

    The Strike job takes a few seconds to run.

When the job is finished, 20 potential models representing the 20 factors extracted are shown in the Results section of the Build QSAR Model panel. The predicted activities for all 20 factors are added to the Project Table under the headers Predicted ActivityX.Y.

With 20 factors, the model fit to the 94 molecules in the training set should have a high R squared, a large F-statistic and a very small P-factor. The standard deviation (Std Dev) should decrease from 1 to about 10 factors and then become somewhat constant while the R squared value should increase continuously as more factors are included, leveling off at about 10 factors. Thus, much of the predictive information is contained in the first 10 factors.

2.6.4 Applying the Model to the Test Set

The true test of any model is to check its predictions against a set of molecules not included during its training. The exercise performed in this section would typically be considered part of model generation and validation, but for the purposes of this tutorial, it will be used to demonstrate the model application step of the Strike workflow.

The first step is to create a test set of molecules. In this example, the test set will be those molecules in the Project Table that were not members of the training set.

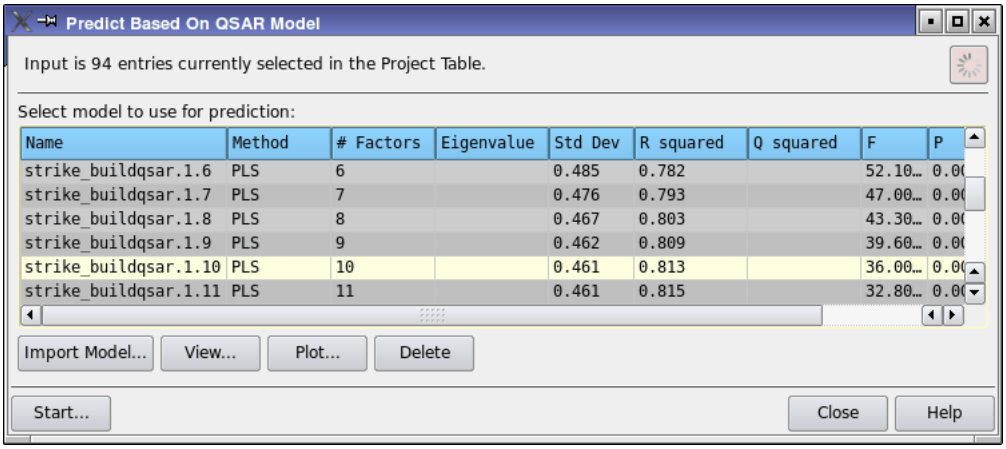

Figure 2.21. Predict based on QSAR Model panel with 10-factor predictor selected

1. In the Project Table, confirm that the training set is selected by examining the Population column.

   If so, skip to the next step. If for any reason the training set is no longer the selected set — for example, if a single entry has been selected instead — you can restore the selection by performing these steps:

   a. Choose Only from the Select menu of the Project Table.

      The Entry Selection panel opens.

   b. In the Properties list, select Population.

   c. Select the option Is defined (any value).

      Only the training set has a defined value (training) in the Population column.

   d. Click the Add button and then the OK button.

   The molecules in the training set, and only those molecules, are now selected. The molecules that were in the training set cannot be part of the test set, so you will invert the selection.

2. In the Project Table, choose Invert from the Select menu.

3. In the Build QSAR Model panel, with the 10-factor predictor 1.10 selected, click Predict.

   The Predict based on QSAR model panel opens showing the model with 20 predictors.

4. Ensure that the 10-factor predictor is selected as shown in Figure 2.21 and click Start.

5. In the Start dialog box, click Start to begin the calculation.

6. When the prediction job is finished, open the Project Table to view the new series of predicted activities for the test set.

Running Strike from Maestro

Before using Strike, molecular data (also referred to as "descriptor data") should be obtained and imported into the Maestro Project Table. This data can be generated using QikProp or other Schrödinger programs. Descriptors for ligands that bind to a receptor can be generated using the Ligand & Structure-Based Descriptors panel. This panel provides an interface to Liaison, Prime, MacroModel (eMBrAcE and `ligparse`), and QikProp to generate descriptors. For more information, see the document *Ligand & Structure-Based Descriptors*. Descriptors generated by external programs or sources may be imported using standard comma-separated value (CSV) format files.

Once the data is incorporated in the Project Table, you can perform statistical analyses and create and use QSAR models using the Strike panels in Maestro.

The Strike interface in Maestro consists of five panels:

- Build QSAR Model—Generate a QSAR model using a training set of molecules selected from the Maestro Project Table, a set of independent descriptors, and a dependent descriptor chosen from those available in the data.
- Predict based on QSAR model—Import a model or select one from the table of generated models, then perform property predictions for molecules that were not part of the training set. Results can be viewed and unsatisfactory models can be deleted from the table.
- Similarity—Determine similarities in descriptor or 2D-structure space. Several distance-based descriptor similarity measures are available. Similarity in 2D-structure space is determined using an atom-pair-based approach.
- Factor Analysis—Perform a principal component analysis and display scores and loadings plots for the principal components.
- Statistics—Display univariate and bivariate statistics.

3.1 The Build QSAR Model Panel

To open the Build QSAR Model panel, choose Build QSAR Model from the Strike submenu of the Applications menu. It may also be useful to open the Project Table containing your molecular data.

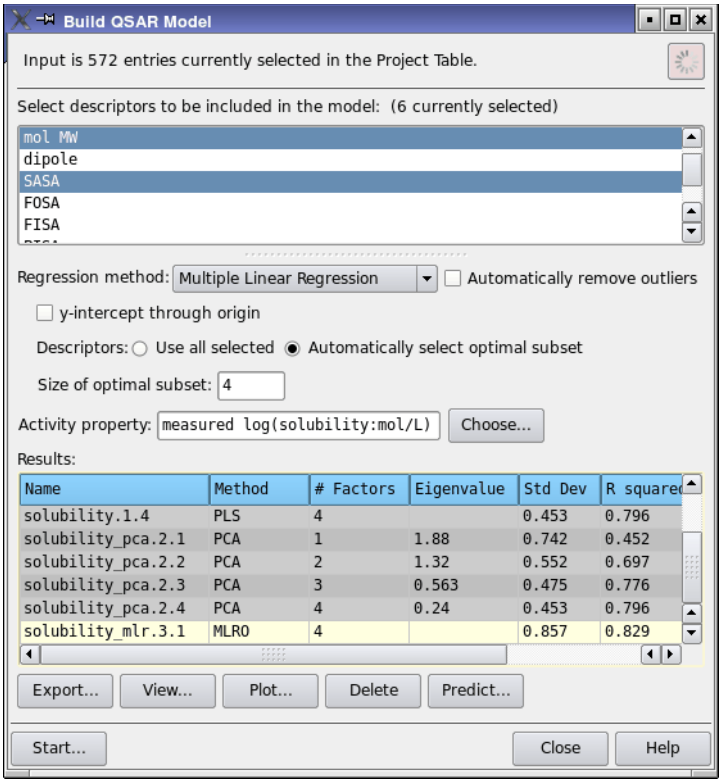

Figure 3.1. The Build QSAR Model panel

3.1.1 Using the Build QSAR Model Panel

Use the panel to generate a QSAR model given a training set of molecules selected from the Project Table, a set of independent descriptors, and a dependent variable for which a prediction will be made. Three regression techniques are available: multiple linear regression (MLR), partial least squares regression (PLS), and principal component analysis (PCA). Certain options are available only for the appropriate regression method.

When you have finished selecting options, click the green Start button to open the Start dialog box for the Build QSAR Model job. Choose the Host machine and the appropriate Username, then click Start.

As the job starts, the pink icon in the upper right corner of the panel turns green and begins to rotate, indicating that a job is in progress, and the job Monitor panel is displayed. The running log for the job appears in the Monitor panel. If you have closed the Monitor panel and want to

reopen it, click the octagon (or choose Monitor Jobs from the Applications menu.) Most model-building jobs take a few seconds to complete.

Once a model has been built, 2D plots of predicted properties versus the dependent descriptor data can be created. Clicking points in a plot brings the molecule or molecules selected into the Maestro 3D Workspace for viewing and manipulation. (For more information about these and other Maestro plots, see Chapter 10 of the *Maestro User Manual*.)

From the Build QSAR Model panel you can proceed directly to the Predict based on QSAR model panel or save the generated models in the project for later use.

3.1.2 Build QSAR Model Panel Features

- Input is *N* entries currently selected in the Project Table

  The entries that are selected (highlighted) in the Project Table will be used to build the QSAR model. The number of entries selected is displayed in the upper portion of the panel.

  Optionally, you can use the Random option in the Select menu of the Project Table to randomly select a specified percentage of either the selected or the total entries. The default is 50% of the selected entries.

- Select descriptors to be included in the model list

  Choose an appropriate set of independent descriptors that is likely to correlate with the dependent descriptor and a regression method by selecting them in the list.

- Regression Method option menu

  The options for regression method are:

  - Partial Least Squares (PLS)

    When this method is selected, the Maximum number of factors option becomes available. The range for Maximum number of factors is from 1 to the number of selected descriptors. The number of molecules selected to be used in building the model must be greater than or equal to the maximum number of factors. For more information about the method, see Section 5.3.2 on page 61.

  - Principal Component Analysis (PCA)

    When this method is selected, the Maximum number of factors option becomes available. The range for Maximum number of factors is from 1 to the number of selected descriptors. The number of molecules selected must be greater than or equal to the number of descriptors chosen. For more information about the method, see Section 5.3.3 on page 61.

- Multiple Linear Regression (MLR)

  When this method is selected, the Descriptors options become available. The number of molecules selected must be greater than or equal to the number of initial descriptors. It is recommended that the number of molecules be at least five times greater than the number of descriptors. For more information about the method, see Section 5.3.4 on page 62.

- Automatically remove outliers

  Select this option to remove outlying molecules before the model is built, using the LOCI algorithm. By default, this option is not selected and outliers are not removed. For samples of 500 members or more, selecting this option will greatly increase the time needed for the model-building job. See Section 5.5 on page 63 for a description of the algorithm. Note that this option does *not* remove outliers based on the model. If you wish to do so, you must run Strike from the command line with the keyword `printMLROutliers` set.

- y-intercept through origin option

  Select this option to force the regression line to pass through the origin.

- Descriptors options (MLR)

  When the regression method selected is MLR, you can choose to Use all selected descriptors (command-line keyword value `MLRS`) or to Automatically select optimal subset of selected descriptors (command-line keyword value `MLRO`).

- Maximum number of factors text box (PLS, PCA)

  When the regression method selected is PLS or PCA, this option is available.

- Size of optimal subset text box (MLR)

  When the regression method selected is MLR, this option is available.

- Activity Property text box and Choose button

  Click Choose to open a list of all the descriptors in the data, from which you can select the property you want the model to predict.

- Results table

  This table lists the models which have been calculated in the current Maestro project. Select a single model to export, view, plot, delete, or use for prediction. Along with the name of each model, the table includes the regression method used, the number of PLS/PCA factors or MLR descriptors, the eigenvalue for PCA models, and standard statistics values.

- Export

  Export the currently selected model to an external file.

- View

  Click View to review the output file of the model-building job for the selected model, which contains all the data needed to completely describe the model.

- Plot

  Generate a Maestro Plot XY plot of the predicted versus experimental activity for the currently selected model.

- Delete

  Delete the currently selected model from the table.

- Predict

  Open the Predict Based On QSAR Model panel with the currently selected model chosen.

3.2 The Predict Based on QSAR Model Panel

The Predict Based On QSAR Model panel allows you to make predictions of molecular properties based on a QSAR/QSPR model. Models to be used for property predictions can be imported from another project or generated in the current project. You must have data for all independent descriptors used in the model.

To open the Predict Based On QSAR Model panel, choose Predict from the Strike submenu of the Applications menu. The Predict Based On QSAR Model panel can also be opened from the Build QSAR Model panel once one or more models have been generated, using the Predict button in the lower portion of the panel.

To make predictions:

1. Select the desired molecules in the Project Table.

   The number of entries selected is displayed at the top of the panel.

2. Select the model to use for the prediction from the table.

   This table contains information about each model in the current project. If you want to view complete information about the model, click View. The output file from the model generation is displayed in a separate panel.

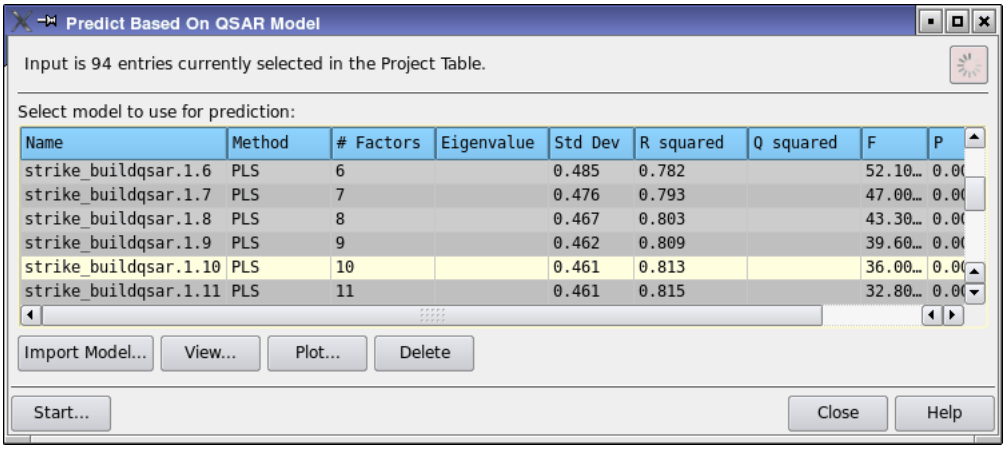

Figure 3.2. The Predict based on QSAR Model panel.

3. Click Start.

   When the job finishes, the predictions are imported into the Project Table and are displayed in the table.

If you want to import a model into the table, click Import Model and navigate to the model (which has the extension `.model`). For example, you can import a model that was exported from the Build QSAR Model panel in an earlier Strike session.

To delete a model from the table, select it and click Delete.

3.3 The Calculate Similarity Panel

The Calculate Similarity panel can be used to determine similarities in descriptor or 2D-structure space. Several distance-based descriptor similarity measures are available. Similarity in 2D-structure space is determined using an atom-pair-based approach.

Similarity, either atom-pair-based or descriptor-based, is calculated with respect to probe molecules. You select the probe molecules by including them in the Workspace. At least one molecule must be included in the Workspace, and at least one entry must be selected in the Project Table, before similarity calculations can proceed.

To open the Calculate similarity panel, choose Similarity from the Strike submenu of the Applications menu.

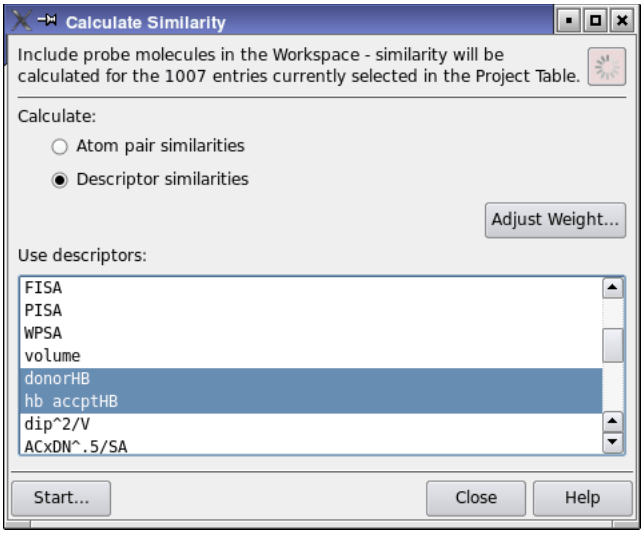

Figure 3.3. The Calculate Similarity panel with descriptor similarities specified.

When you open the panel, the number of selected entries is displayed at the top of the panel. There are two options for the type of similarity to be calculated:

- Atom pair similarities—A similarity property will be created for each selected entry in the Project Table, based on the similarity of the atoms in the structure.
- Descriptor similarities—Similarity in terms of properties will be calculated for the selected Project Table entries in terms of the properties chosen from the Use Descriptors list. When you choose this option, the Use Descriptors list becomes available. and you can choose the properties you want to include in the similarity calculation. If you want to adjust the weights of the chosen descriptors in the evaluation of the similarity, click Adjust Weight and set the desired weights in the Adjust Weight dialog box. The default weight is 1.0 for all descriptors.

3.4 The Factor Analysis Panel

The Factor Analysis panel provides tools for performing principal component analyses of various sets of descriptors taken from a set of entries that is selected in the Project Table. Once an analysis is done, you can plot the scores and the loadings for selected pairs of principal components, or make use of the principal component decomposition for further model generation. For background on principal component analysis, see Section 5.3.3 on page 61.

The scores plot shows clustering of molecules from plotting one set of principal components against another. Each point on the plot represents a molecule. Molecules with similar data will

be found together. Often it is possible to visually cluster molecules on the basis of the scores plot.

The loadings plot (also known as a weights plot) shows the influence of individual variables (descriptors) on the principal components. Each point on the plot represents a variable from the input data. Larger values for a given point indicates it has more weight in a given principal component. Often it is possible to visually cluster the effect of variables on the principal components.

By looking at a score and loading plot side-by-side it is possible to identify relationships between molecules and variables. For instance, if a number of molecules cluster with large values of the first principal component (PC1) in the scores plot, you can use the loadings plot to determine which variables have the most weight in PC1. These molecules will then cluster nicely on the basis of those variables.

To generate the principal components:

1. Select the entries in the Project Table for which you want to perform the analysis.
2. Choose Factor Analysis from the Strike submenu of the Applications menu.

   The Factor Analysis panel opens. The descriptor list is populated with the properties available in the Project Table. The visualization tools are hidden until an analysis job is run.
3. Select a set of descriptors from the descriptor list.

   You can use shift-click and control-click in the usual way to select multiple list items.

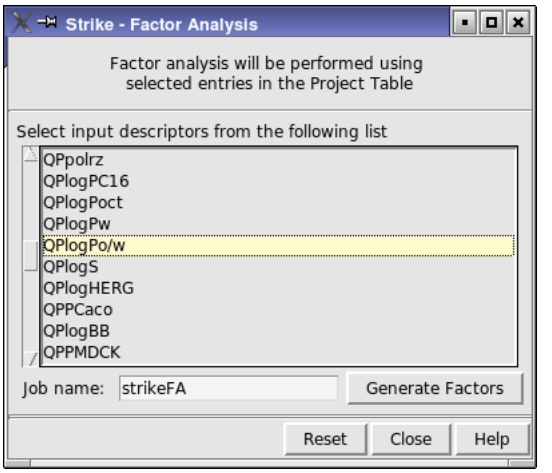

Figure 3.4. The Factor Analysis panel

4. Name the job in the Job name text box.

   It is a good idea to choose a name that has some relation to the set of descriptors selected, since the name is displayed in the Visualize data from option menu once the first analysis job finishes.

5. Click Generate Factors.

   After a short time, the job finishes, and the visualization tools are displayed in the lower part of the panel.

6. Repeat Step 3 through Step 5 for each set of descriptors that you want to analyze.

To display the results of an analysis:

1. Select the job from the Visualize data from option menu.
2. Select the principal components for the *x* and *y* axes for the desired plot types.

   By default, component 1 and component 2 are selected.

3. Click the Plot Scores button or the Plot Loadings button.

   The Plot XY panel opens with the selected data plotted. You do not need to close this panel to display another plot: it will simply be added to the set of plots in the panel.

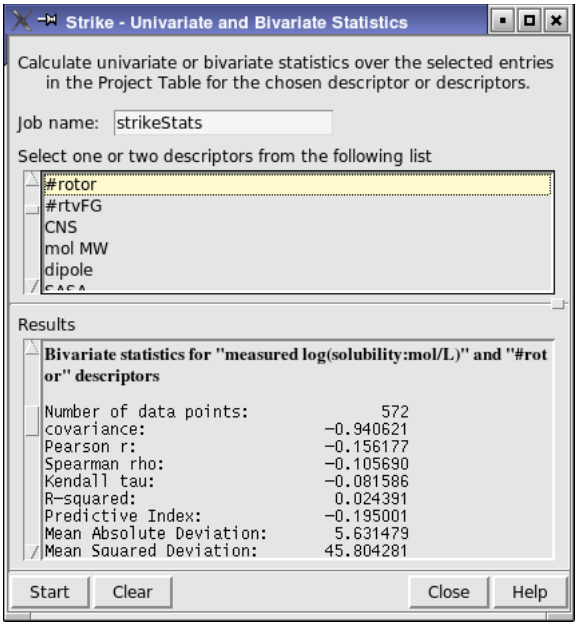

Figure 3.5. The Univariate and Bivariate Statistics panel with bivariate statistics

3.5 The Univariate and Bivariate Statistics Panel

The Univariate and Bivariate Statistics panel enables you to calculate univariate statistics or bivariate statistics over the entries that are selected in the Project Table. To display univariate statistics for a particular descriptor, select the descriptor from the list of descriptors, then click Start. To display univariate and bivariate statistics for a particular descriptor, select two descriptors from the list of descriptors (click the first, control-click the second), then click Start. The results are displayed in the Results text area. The statistics given are described in Section 5.1 on page 55 and Section 5.2 on page 58.

Running Strike from the Command Line

Strike can also be run using the `strike` command. This chapter lists keywords for the input file and gives two examples of command blocks that can be used in input files.

4.1 Usage Summary

`$SCHRODINGER/strike [`*options*`]` *inputfile*

inputfile	The `strike` input script file containing the commands to be performed.
`-HOST` *host*	Run job on a remote host.
`-LOCAL`	Run the job in the current directory, rather than in a temporary scratch directory.
`-WAIT`	Do not return until the job completes.
`-NICE`	Run the job at reduced priority.
`-HELP`	Print this message and exit.

4.2 Input File Examples

The `strike` input script consists of blocks of commands, each consisting of a series of *keyword=value* pairs and terminated by a line beginning with `#`. The termination line beginning with `#` is mandatory, even if there is only one block of data in the input script. Each command block is then executed sequentially. Comment lines must begin with `!!`. Two examples of command blocks are given below.

```
!! Section to train a model
dataFile=input_files/strike_mlro_fit.csv
runMode=train
model=MLRO
activityLabel=activity
numOptDescript = 4
# End of Section 1

!! Section to use a previously created model
runMode=test
dataFile=input_files/strike_pls_fit.csv
modelFile=input_files/strike_pls_fit.model
# End of Section 2
```

4.3 Input File Keywords

The following *keyword=value* pairs are accepted input for `strike`. Boolean values must be expressed as `yes` or `no`.

4.3.1 Mode Selection

All Strike jobs must use the `runMode` keyword with one of these values.

Value	Description
`train`	Generate a QSAR model.
`test`	Predict properties using a QSAR model.
`simil`	Run a descriptor similarity calculation.
`apsimil`	Run an atom-pair (2D structure space) similarity calculation.
`stats`	Generate statistics.
`factorGen`	Extract factors from PCA model.
`factorRed`	Reduce data using extracted PCA factors.
`factorExp`	Expand reduced data using extracted PCA factors.

4.3.2 File Specification Commands

Keyword	Value	Description/Relevant Job Types
`dataFile`	*datafilename*	Keyword required for all jobs except atom-pair similarity (`apsimil`). File must be in CSV or Maestro format.
`outputFile`	*outputfilename*	Default is *jobname*`.out`. All jobs.
`modelFile`	*modelfilename*	Default is *jobname*`.model`. File containing QSAR model. QSAR jobs (`train` and `test`) and factor reduction jobs (`factorGen`, `factorRed`, `factorExp`).
`csvFile`	*csvoutfilename*	Default is *jobname*`.csv`. Output file containing all data used and generated in current command block. All jobs except `apsimil`.
`plotFile`	*qsaroutfilename*	File containing output from `train` with predicted vs. dependent data. QSAR jobs.
`apPredFile`	*filename*	File of molecules whose similarity to the probes are to be determined. `apsimil` jobs.
`apActivesFile`	*activesfilename*	File of probe molecules. `apsimil` jobs.
`apInactivesFile`	*inactivesfilename*	File of decoy molecules. `apsimil` jobs.
`apWeightsFile`	*weightsfilename*	Weights file. When generated, default is *jobname*`.csv`. `apsimil` jobs.

4.3.3 Alternative Naming Convention Commands

Keyword	Value	Description/Relevant Job Types
`modelTitle`	*modelname*	Alternative title for QSAR model generation. Otherwise defaults to *jobname*.
`baseName`	*basename*	Alternative basename for all jobs. All output files will be *basename*`.`*, and `modelTitle` will default to *basename*.

4.3.4 Commands for Reading/Writing .csv Files

Keyword	Value	Description/Relevant Job Types
`delim`	*string*	Delimiter character for reading `.csv` file. All jobs except `apsimil`.
`includeColumns`	*X:Y, Z* column numbers or column labels	X, Y, Z can be numbers or labels (headers). Use colon for ranges, For example, `includeColumns=2:6,9,15` includes columns 2-6, 9, and 15 from the input file. All jobs except `apsimil`.
`excludeColumns`	*X:Y, Z* column numbers or column labels	X, Y, Z can be numbers or labels (e.g., labels: `excludeColumns=IP(ev):QPlogKhsa` Properties in `.csv` file between IP(ev) and QPlogKhsa, inclusive, will not be used). All jobs except `apsimil`.
`includeRows`	*X:Y, Z* row numbers or row labels	Include molecules (rows) specified. All jobs except `apsimil`.
`excludeRows`	*X:Y, Z* row numbers or row labels	Exclude molecules (rows) specified (e.g., `excludeRows=25` excludes molecule 25). All jobs except `apsimil`.
`activityColumn`	*integer*	Identify dependent property by column number. Build QSAR model jobs.
`activityLabel`	*label*	Identify dependent property by column label. Build QSAR model jobs.
`rowHeaderColumn`	*integer*	Set the column in the `.csv` file that contains row labels, by column numbering beginning at 1. For all jobs if needed.

Keyword	Value	Description/Relevant Job Types
rowHeaderLabel	*label*	Set the column in the .csv file that contains row labels by column label. For all jobs if needed.
descriptorWeightRow	*integer*	Set by **number** the row that contains the weight for each descriptor. For descriptor similarity jobs.
descriptorWeightLabel	*label*	Set by **label** the row that contains the weight for each descriptor. For descriptor similarity jobs.

4.3.5 Commands for Build QSAR Model (train) Jobs

Keyword	Value	Description/Relevant Job Types
model	PLS PCA MLRS MLRO NNET	Specify type of regression to be employed.
autoScale	yes no	Set whether data is to be converted to a common scale. Default is yes.
maxFactors	*integer*	Maximum number of factors to return. PLS, PCA.
numOptDescript	*integer*	Number of descriptors to be retained, determined by optimization. MLRO.
removeOutliers	no yes	Run prior to importing data into model building. Compare relative densities in descriptor space for included molecules to predict outliers. Recommended for sample size < 500 only. Default is no.
printMLROutliers	no yes	Set to yes to output possible outliers with respect to the MLR model. Default is no. MLRS, MLRO.
MLROutlierCutoff	*integer*	Integer from 0 to 5 giving the number of MLR outlier tests that need to fail before a data point is identified as a possible model outlier. Default is 4. MLRS, MLRO.
lgoPercent	double	For leave-group-out (LGO) validation, percentage of fitting set to use as test set for each regression. Default is 5.0%
lgoCycles	*integer*	Number of cycles of LGO validation to perform. Default is 10.
RandCycles	*integer*	Number of randomization cycles to perform. Default is 10 times the number of independent descriptors. MLRS, MLRO, PLS, and PCA.
supYintercept	no yes	Suppress inclusion of the *y* intercept as a dependent variable for regression generation. The default is no, which means that the *y* intercept is included. MLRS, MLRO.

Keyword	Value	Description/Relevant Job Types
nnetNumUnitsInHidden Layer	*integer*	Number of units in the hidden layer. NNET.
nnetCrossValPer	*integer*	Percent of input data to be kept in the cross validation set. Default is 5%. NNET.
nnetExtValPer	*integer*	Percent of input data to be kept in the external validation set. Default is 10%. NNET.
nnetNumTrainCycles	*integer*	Number of training cycles for each neural network. Default is 200. NNET.
nnetNumNetworks	*integer*	Number of neural networks to train of which the best nnetumNetworksEnsem will be selected to create an ensemble neural network that is presented to the user. Default is 20. NNET.
nnetNumNetworksEnsem	*integer*	Number of the best neural networks to use in generating an ensemble neural network that is presented to the user. Default is 5. NNET.

4.3.6 Commands for Atom-Pair Similarity (apsimil) Jobs

Keyword	Value	Description
apPredFormat	mae sdf	Format of apPredFile.
apActivesFormat	mae sdf	Format of apActivesFile.
apInactivesFormat	mae sdf	Format of apInactivesFile.
probes	*X:Y, Z*	Specify probe molecules. *X, Y, Z* can be molecule numbers or, if rowHeaderColumn is defined, molecule titles.
inactivePercent	*nn.n*	Percentage of inactives, e.g., for 99%: inactivePercent=99.0
readWeights	yes no	Read weights from apWeightsFile.
genWeights	yes no	Generate weights in apWeightsFile.
normalize	range z-score none	Specify normalization approach. Default (range) normalizes data to 0.0 - 1.0 scale; required for Tanimoto coefficient calculation. Specify z-score to scale data in standard deviation units. Specify none to perform no normalization.

4.3.7 Commands for Factor Reduction Jobs

Keyword	Value	Description
facRedAuto	yes no	Determines if the factors are to be generated using scaled (yes) or unscaled (no) input data.
facRedNumFactors	*n*	Number of factors to generate. Range is from 0 to the number of input data columns.

4.3.8 Other Commands

Keyword	Value	Description/Relevant Job Types
enrich	Euclidean Euclidean_sq Tanimoto Manhattan	Calculate enrichment factors for extracting probe molecules from the entire data set, using the specified similarity measure. For simil and apsimil jobs.
stats	*label*	Calculate univariate statistics for the descriptor *label*. Any job.
	label1 , *label2*	Calculate bivariate statistics for the descriptors *label1* and *label2*. Any job.

4.3.9 Keyword Requirements for Various Job Types

Table 4.1. Minimum Strike Keywords by Job Type

Job Type	Keywords Required	Comments
Build QSAR model	Keywords depend on chosen model.	Column containing dependent data can be specified by column number (activityColumn) *or* by column label (activityLabel).
Build QSAR model Partial Least Squares	runMode=train model=PLS dataFile activityColumn *or* activityLabel maxFactors	
Build QSAR model Principal Component Analysis	runMode=train model=PCA dataFile activityColumn *or* activityLabel maxFactors	

Table 4.1. Minimum Strike Keywords by Job Type (Continued)

Job Type	Keywords Required	Comments

Build QSAR model Multiple Linear Regression Analysis	`runMode=train`
	`model=MLRS`
	`dataFile`
	`activityColumn` *or*
	`activityLabel`	

Build QSAR model MLRS with optimum number of descriptors	`runMode=train`
	`model=MLRO`
	`dataFile`
	`activityColumn` *or*
	`activityLabel`
	`numOptDescript`	

Build QSAR model Neural Network	`runMode=train`
	`model=NNET`
	`dataFile`
	`activityColumn` *or*
	`activityLabel`
	`nnetNumUnitsInHidden Layer`	

Validation or prediction using any model	`runMode=test`
	`dataFile`
	`modelFile`
		Model is entirely specified in `modelFile`.

Descriptor similarity calculation	`runMode=simil`
	`dataFile`
	`probes`
	(`rowHeaderColumn` *or*
	`rowHeaderLabel`)
		One of the keywords in parentheses needed to specify probe molecules if probes not specified by molecule number.

Atom-pair similarity calculation. No weights	`runMode=apsimil`
	`apActivesFile`
	`apActivesFormat`
	`apPredFile`
	`apPredFormat`	

Atom-pair similarity. Weights are generated and used in same command block	`runMode=apsimil`
	`apActivesFile`
	`apActivesFormat`
	`apPredFile`
	`apPredFormat`
	`apInactivesFile`
	`apInactivesFormat`
	`inactivePercent`
	`genWeights`	

Table 4.1. Minimum Strike Keywords by Job Type (Continued)

Job Type	Keywords Required	Comments

Atom-pair similarity using weights that were generated previously	`runMode=apsimil`
	`apActivesFile`
	`apActivesFormat`
	`apPredFile`
	`apPredFormat`
	`readWeights`
	`apWeightsFile`	

PCA factor generation	`model=PCA`
	`runMode=factorGen`
	`facRedNumFactors`
	`facRedAuto=yes`
	`dataFile`
		Also generates reduced data set for the input.

PCA factor reduction	`model=PCA`
	`runMode=factorRed`
	`facRedNumFactors`
	`facRedAuto=yes`
	`dataFile`
	`modelFile`
		`modelFile` must be output of a factor generation job. `dataFile` must contain the same descriptors as used in the factor generation, but need not contain data for the same structures.

PCA factor expansion	`model=PCA`
	`runMode=factorExp`
	`facRedNumFactors`
	`facRedAuto=yes`
	`dataFile`
	`modelFile`
		`modelFile` must be output of a factor generation job. `dataFile` must be a file with reduced factors from a factor generation or reduction job.

Statistical Definitions and Methods

This chapter defines statistical quantities, algorithms, and regression methods used in Strike.

5.1 Univariate Statistics

This section defines some symbols, definitions, and equations relating to the statistics of a single variable.

5.1.1 Symbols

N

Number of data points (observations) in a sample. There is no hard limit on sample size (number of molecules) in Strike, but for large samples (millions of molecules) practical issues such as system memory limitations may apply.

x_i —value of data point i in a sample of variable x

$\bar{x}$ —mean of variable x

5.1.2 Mean, Median, and Mode

These statistics describe the "central tendency" of a variable.

Mean

The *mean* of variable x is defined by Equation (1).

$$\bar{x} = \sum_{i}^{N} x_i / N \tag{1}$$

Median

- For an even number of data points, the *median* is the mean of the middle-most two values in the ordered sample.
- For an odd number of data points, the *median* is the middle-most value in the ordered sample.

One-half of the ranked values for a variable will lie above and one-half below the value of the median.

Mode

The *mode* is the value of a variable that occurs with the greatest frequency in a sample. If more than one value shares the highest frequency of occurrence, the term "mode" is not applicable.

5.1.3 Variance and Deviation

The statistical quantities defined in this section are measures of the spread of values in a sample about the mean.

Variance

The *variance* is defined as:

$$\sigma^2 = \sum_{i}^{N} (x_i - \bar{x}) / (N - 1) \tag{2}$$

If $\bar{x}$ is known or if one is examining a complete population, the N–1 term reverts to N. Strike calculates all variances assuming a sample, i.e. using N–1, which is suitable for the vast majority of cases. The *mean squared deviation*, also defined in this section, reports the variance for a population.

Standard Deviation

The *standard deviation* is the square root of the variance, defined in Equation (2).

If $\bar{x}$ is known or if one is examining a complete population, the N–1 term reverts to N. Strike calculates all standard deviations assuming a sample, i.e. using N–1, which is suitable for the vast majority of cases. The *root mean squared deviation*, also defined in this section, reports the standard deviation for a population.

The standard deviation measures the spread of values about the mean. If the sample exhibits a normal distribution, then 68.3% of values will lie within 1σ from the mean, 95.4% of values will lie within 2σ of the mean, and 99.7% of values will lie within 3σ of the mean.

Mean Absolute Deviation

$$\mathrm{MAD} = \sum_{i}^{N} \left| x_i - \bar{x} \right| / (N - 1) \tag{3}$$

Mean Squared Deviation

$$\mathrm{MSD} = \sum_{i}^{N} (x_i - \bar{x})^2 / N \tag{4}$$

If $\bar{x}$ is known or if one is examining a complete population, the *mean squared deviation* reports the variance for the complete population.

Root Mean Squared Deviation

$$\mathrm{RMSD} = \sqrt{\sum_{i}^{N} (x_i - \bar{x})^2 / N} \tag{5}$$

If $\bar{x}$ is known or if one is examining a complete population, the *root mean squared deviation* reports the standard deviation for the complete population.

5.1.4 Skewness and Kurtosis

These statistics are measures of the extent to which a sample differs from a normal distribution. Both have a value of zero for a normal distribution.

Skewness

$$\mu = \frac{\sum_{i}^{N} (x_i - \bar{x})^3 / N}{\mathrm{RMSD}^3} \tag{6}$$

Strike calculates the Fisher Skewness for a sample. Normal distributions have a skewness of zero as they are perfectly symmetrical about the mean. A positive value of the skewness indicates, relative to a normal distribution, that the sample being examined is asymmetric and skews towards larger values, i.e. has a larger tail to the right. A negative value of the skewness indicates, relative to a normal distribution, that the sample being examined skews toward smaller values, i.e. has a larger tail to the left. A significant skewness value indicates that the sample does not have a normal distribution.

Kurtosis

$$\text{kurtosis} = \frac{\sum_{i}^{N}(x_i - \bar{x})^4 / N}{\text{RMSD}^4} - 3 \tag{7}$$

Strike calculates the excess kurtosis using the formula of Snedecor and Cochran. The kurtosis for a normal distribution is three. By subtracting three, Strike reports the excess kurtosis where a normal distribution has a kurtosis of zero. A positive kurtosis indicates the distribution is strongly peaked about the mean while a negative kurtosis indicates the distribution is flat. A significant kurtosis value indicates the sample does not have a normal distribution.

5.2 Bivariate Statistics: Covariance and Correlation

The statistics in this section describe the relationship between two variables in terms of covariance and correlation. These statistics are also applied to pairs of variables in models with multiple independent variables.

Correlation coefficient

See *Pearson r* or *r-squared*

Correlation matrix

The *correlation matrix* generates the Pearson *r* values for the half matrix of all pairs of selected variables.

Covariance

$$\text{cov}(x, y) = \frac{\sum_{i}^{N}(x_i - \bar{x})(y_i - \bar{y})}{N - 1} \tag{8}$$

The covariance measures the extent to which two variables vary together. A positive value of the covariance indicates that larger than average values of one variable tend to be paired with larger than average values of the second variable. A negative value of the covariance indicates that larger than average values of one variable tend to be paired with smaller than average values of the second variable. A zero covariance indicates the two variables vary independently from one another. The covariance is dependent on the magnitude of the variables involved and is most useful when the variables have the same magnitude.

For a scatter plot of x and y the covariance measures how close the scatter is to a line. A negative covariance indicates a downward sloping line to the right, a positive covariance indicating an upward sloping line to the right, and a zero covariance indicating the best line lies along the horizontal axis.

Pearson r

$$r = \frac{\mathrm{cov}(x, y)}{\sigma_x \sigma_y} \tag{9}$$

The *Pearson r* is a correlation coefficient that determines the extent that two variables are proportional to one another. In other words, the Pearson *r* provides a measure of linear association between variables. Calculated Pearson *r* values lie on a scale from -1.0 to +1.0 with negative values indicating the best least-squares line between variables x and y is downward sloping to the right and positive values indicating the best line is upward sloping to the right. A value of zero indicates no correlation between the two variables. The Pearson *r* is independent of the magnitude of variables (unlike the covariance). Note that *R* is sometimes used instead of *r*.

R-squared

In Strike, *r-squared* is the square of the Pearson *r* correlation coefficient. Its value ranges from 0.0 to 1.0 with a value of zero indicating the two variables have no correlation and a value of one indicating the variables are perfectly correlated. Like the Pearson *r*, the *r*-squared is independent of the magnitude of the two variables.

Spearman rho

The *Spearman rho* is a rank-order correlation coefficient. It measures the proportion of variability accounted for between two variables using the ranking of the data rather than the data values themselves. The Spearman rho is interpreted in an identical fashion to the Pearson r statistic.

Ties in ranking (data points with the same value) are given the mean rank of the tied observations. i.e. if three points are identified as having equal values with ranks of 5, 6, 7, and 8 in the sample, the average rank assigned to all four would be 6.5. In the definitions below, $\mathrm{rank}(x_i)$ is the rank of the point x_i, $\mathrm{ties}(x_i)$ is the number of times the value x_i occurs, and in $\varepsilon(x)$ the sum is over the number of tied values.

$$D = \sum_{i}^{N} [\mathrm{rank}(x_i) - \mathrm{rank}(y_i)]^2 \tag{9a}$$

$$\varepsilon(x) = \sum_i \text{ties}(x_i)^3 - \text{ties}(x_i) \tag{9b}$$

$$\rho = \frac{1 - [6D + (\varepsilon(x) + \varepsilon(y))/2]/(N^3 - N)}{\sqrt{1 - \varepsilon(x)/(N^3 - N)}\sqrt{1 - \varepsilon(y)/(N^3 - N)}} \tag{9c}$$

Kendall tau

The *Kendall tau* is a rank-order correlation coefficient. It measures the proportion of variability accounted for between two variables using the ranking of the data rather than the data values themselves. The Kendall tau is interpreted in an identical fashion to the Pearson *r* statistic. It is defined in Equation (10), where:

P is the number of concordant pairs of ranks

Q is the number of discordant pairs of ranks

Y_0 is the number of ties in the ranks of two x's

X_0 is the number of ties in the ranks of two y's

$$\tau_b = \frac{P - Q}{\sqrt{P + Q + X_0}\sqrt{P + Q + Y_0}} \tag{10}$$

To calculate the Kendall tau the half matrix of data pairs is analyzed, i.e. (x_i,y_i) and (x_j,y_j) are compared for all i and j pairs. Each pair that shows the same rank order between the two data sets is counted as concordant. Each pair that shows a different rank order between the two data sets is counted as discordant. The rank order can be determined by the following expression:

$$\begin{aligned} (\text{rank}(x_i) - \text{rank}(x_j))(\text{rank}(y_i) - \text{rank}(y_j)) &> 0 \quad \text{concordant} \\ &< 0 \quad \text{discordant} \\ &= 0 \quad \text{tie in } x \text{ or } y \end{aligned} \tag{11}$$

Ties are counted in the Y_0 and X_0 variables.

5.3 Model-Building Methods

This section defines terms and methods used in building QSAR/QSPR models. It briefly introduces the three regression methods available for model-building in Strike: partial least squares, principal component analysis, and multiple linear regression.

5.3.1 Independent and Dependent Variables

Dependent variable

The *dependent variable* (or *response variable*) is the variable that is being fitted to in a regression model. It is referred to as dependent as it is assumed that its values are dependent on the values of independent variables that will be used to generate the predictive model. In Strike, this variable is also referred to as the dependent descriptor or the activity property.

Independent variables

The *independent variables* are the variables that are being used to fit a regression to a dependent variable in partial least squares, principal component analysis, or multiple linear regression. They are referred to as independent as their values are assumed not to depend on the values of the dependent variable. In Strike, the term *independent descriptors* is often used.

5.3.2 Partial Least Squares

The *partial least squares* (PLS) method generates linear equations that describe the relationship between a number of factors derived from a set of independent descriptors and a dependent descriptor. The PLS procedure works by extracting successive linear combinations of the factors (also called components or latent vectors), which explain independent and dependent variations. In particular, the method of partial least squares balances these objectives, seeking factors that explain both response variation and predictor variation.

Partial least squares is particularly valuable because it can be applied in cases where the number of independent descriptors is greater than the number of molecules.

Partial least squares is similar to *principal component analysis*, but the goals of the two methods in extracting factors differ. In PLS one is concerned with the variance in both the dependent and independent descriptors, while in PCA one is trying to explain the maximum variance possible in only the dependent descriptors.

While it is possible to use PLS to generate models to fit multiple dependent variables, Strike is limited to fitting a single dependent variable.

5.3.3 Principal Component Analysis

Principal component analysis (PCA) transforms a number of independent variables into a number of uncorrelated factors that explain the variance of the dependent variable. The first factor accounts for as much of the variability in the data as possible, and each succeeding factor accounts for as much of the remaining variability as possible. The eigenvalues of the covariance matrix from PCA indicate the portion of the total variance accounted for by each

factor, where the total variance is generally defined as equal to the number of independent variables.

Principal component analysis can be applied in cases where the number of independent descriptors is greater than the number of molecules.

Principal component analysis is similar to partial least squares, but it focuses on explaining the maximum variance possible in only the dependent descriptors, while PLS considers the variance in both the dependent and independent descriptors.

While it is possible to use PCA to generate models to fit multiple dependent variables, Strike is limited to fitting a single dependent variable.

5.3.4 Multiple Linear Regression

Multiple linear regression (MLR) generates linear equations that describe the relationship between a set of independent descriptors and a dependent descriptor. Strike may only be used to fit a single dependent descriptor. As used in Strike, MLR fits a straight line to the dependent descriptor using the following linear relationship:

$$P_j = \sum_i c_i \chi_{ij} + c_0 \tag{12}$$

In the above equation, P_j is the property or activity that is to be predicted for each molecule j, the c_i values are the regression coefficients, χ_{ij} is the ith independent property for molecule j, and c_0 is a constant. Values of the coefficients and c_0 are fitted to give P_j values that reproduce the dependent value for the jth molecule.

In general, when fitting data using MLR it is advisable to use a data set with at least five times as many molecules as there are independent descriptors.

5.4 Model Analysis and Validation

The statistics in this section are used to analyze and validate QSAR/QSPR models built using regression techniques.

Cross validation or leave-*n*-out validation

Cross validation tests how dependent a generated regression is on the samples used to generate the regression. In leave-group-out (LGO) or leave-*n*-out cross validation, the original set of samples is divided into n subsets. Then, n regressions are generated, each time omitting a different subset. Each of the n regressions is then used to predict the expected dependent value for the molecules in the omitted subset. In n regressions all molecules will have had their

dependent value predicted and the r-squared from comparing the predicted dependent values against the true dependent values is referred to as the *q-squared*. To reduce the dependence of cross validation on the composition of the subsets randomly generated, the cycle is repeated *c* times. The mean of the *c* values of q-squared is reported by Strike. A q-squared value that deviates significantly from the r-squared for a regression generally indicates that the regression is overly dependent on the set of molecules included in the training set and may not have the desired predictive power.

By default, Strike uses a subset size (`lgoPercent`) of 5% of the sample, giving *n* subsets = 20, and a number of cycles *c* (`lgoCycles`) = 10. When Strike is run from the command line, the `lgoPercent` and `lgoCycles` keywords can be used to specify non-default values. See Section 4.3.5 on page 50.

F-statistic

The *F-statistic* is used in regression analysis to determine if the variances between the means of two populations are significantly different. In other word, the F-statistic provides an indication of the lack of fit of the data to the estimated values of the regression. A strong relationship between two variables gives a high F-ratio.

Leave-*n*-out validation

See *cross validation*.

P-value

The *p-value* is the probability that the regression was obtained not from correlations between the dependent and independent variables, but instead by chance. Generally p-values of < 0.05, which indicate a 1 in 20 probability that the regression was obtained by chance, are considered statistically significant.

Q-squared

The *q-squared* is the r-squared determined by comparing the dependent variable against predictions made using a model. See *cross validation* for details.

5.5 Outlier Detection

Local Correlation Integral Outlier (LOCI) Detection

The LOCI outlier detection methodology uses a density-based approach to identifying outliers within a sample. It works by comparing the density of points surrounding a given point with the densities of the surrounding neighbor points. Significant differences in densities lead to the identification of outliers. It provides an automatic, data dictated cut-off to identify outliers

without the need for user input. This method does not suffer from either the local density problem or the multi-granularity problem. It should be noted that our implementation of the LOCI algorithm does not scale well for larger sample sizes, and as such should only be used on samples of less than about 500 members.

Outliers Identified in Multiple Linear Regression

A second outlier detection method is used in conjunction with multiple linear regression (MLR). In this case, the model is run first and a set of five statistical tests on the final MLR model is run to identify the outliers. The five tests are:

- Standardized residual
- Studentized residual
- Leverage
- DFFITS
- Cook's test

A molecule must be flagged by at least 2 of the five MLR outlier tests by default to be flagged for a molecule before listing it as being a possible outlier.

the keyword `printMLROutliers` is used

5.6 Similarity Statistics

The statistics defined in this section are measures of similarity.

5.6.1 Atom-Pair Similarity

m_{AB} is the total number of unique atom pair types found on molecules A and B

freq_k^A is the number of times atom pair type k was found on molecule A

w_i is the weight for atom pair type k

$$\text{sim}_{AB} = \frac{\sum_{k}^{m_{AB}} w_k \min(\text{freq}_k^A, \text{freq}_k^B)}{0.5 \sum_{k}^{m_{AB}} w_k (\text{freq}_k^A + \text{freq}_k^B)} \quad (13)$$

To calculate the atom-pair similarity of two molecules, a set of atom-pair types is developed for each molecule. The atom-pair types are determined using the hydrogen-suppressed graph of the chemical structure and combining a simple atom typing scheme with the shortest path

distances to arrive at the set of atom-pair types in the form, $type_i$-d_{ij}-$type_j$. The number of atom-pair types the two molecules share will determine their atom-pair similarities with 0.0 indicating no similarity and 1.0 indicating all atom pairs of the two molecules are shared. The atom-pair weights are all 1.0 by default though they may be fitted to bias important atom pairs. Weight fitting and application in Strike can only be done from the command line. See Section 4.3.6 on page 51.

5.6.2 Similarity Measures in Descriptor Space

The four quantities defined here are measures of distance in descriptor space.

Manhattan Distance

$$\text{dist} = \sum_i w_i \left| x_i - x_i^{\text{probe}} \right| \tag{14}$$

The Manhattan distance metric, also known as the city-block distance, is a measure of the sum of geometric distances between points measured along axes at right angles. The distance being measured is summed over all variables. Put another way, the distance is calculated between all descriptors for a molecule and the probe value for each of those descriptors. For descriptor similarities, the probe value for each descriptor is the mean of the values of each probe molecule for that descriptor. Different weights for each descriptor, w_i, may be included only in command-line Strike calculations, otherwise all weights have the value of one. A value of zero indicates the probe molecule and test molecules are identical.

Euclidian Squared Distance

$$\text{dist} = \sum_i w_i (x_i - x_i^{\text{probe}})^2 \tag{15}$$

The Euclidean squared distance metric is a measure of the sum of geometric distances between points. The distance being measured is summed over all variables. Put another way, the distance is calculated between all descriptors for a molecule and the probe value for each of those descriptors. For descriptor similarities, the probe value for each descriptor is the mean of the values of each probe molecule for that descriptor. Different weights for each descriptor, w_i, may be included only in backend Strike calculations, otherwise all weights have the value of one. A value of zero indicates the probe molecule and test molecules are identical.

Euclidian Distance

The Euclidean distance is the square root of the expression in Equation (15).

Tanimoto Similarity

$$\text{dist} = \frac{\sum_i x_i x_i^{\text{probe}}}{\sum_i x_i x_i + \sum_i x_i^{\text{probe}} x_i^{\text{probe}} - \sum_i x_i x_i^{\text{probe}}} \tag{16}$$

The Tanimoto distance metric is a normalized measure of the similarity in descriptor space between a test molecule and a probe molecule. Similarities lie between one and zero with a value of one indicating identical molecules and a value of zero indicating completely dissimilar molecules.

Getting Help

Schrödinger software is distributed with documentation in PDF format. If the documentation is not installed in `$SCHRODINGER/docs` on a computer that you have access to, you should install it or ask your system administrator to install it.

For help installing and setting up licenses for Schrödinger software and installing documentation, see the *Installation Guide*. For information on running jobs, see the *Job Control Guide*.

Maestro has automatic, context-sensitive help (Auto-Help and Balloon Help, or tooltips), and an online help system. To get help, follow the steps below.

- Check the Auto-Help text box, which is located at the foot of the main window. If help is available for the task you are performing, it is automatically displayed there. Auto-Help contains a single line of information. For more detailed information, use the online help.
- If you want information about a GUI element, such as a button or option, there may be Balloon Help for the item. Pause the cursor over the element. If the Balloon Help does not appear, check that Show Balloon Help is selected in the Maestro menu of the main window. If there is Balloon Help for the element, it appears within a few seconds.
- For information about a panel or the tab that is displayed in a panel, click the Help button in the panel, or press F1. The help topic is displayed in your browser.
- For other information in the online help, open the default help topic by choosing Online Help from the Help menu on the main menu bar or by pressing CTRL+H. This topic is displayed in your browser. You can navigate to topics in the navigation bar.

The Help menu also provides access to the manuals (including a full text search), the FAQ pages, the New Features pages, and several other topics.

If you do not find the information you need in the Maestro help system, check the following sources:

- *Maestro User Manual*, for detailed information on using Maestro
- *Maestro Command Reference Manual*, for information on Maestro commands
- *Maestro Overview*, for an overview of the main features of Maestro
- *Maestro Tutorial*, for a tutorial introduction to basic Maestro features
- Strike Frequently Asked Questions pages, at https://www.schrodinger.com/Strike_FAQ.html
- Known Issues pages, available on the Support Center.

The manuals are also available in PDF format from the Schrödinger Support Center. Local copies of the FAQs and Known Issues pages can be viewed by opening the file `Suite_2009_Index.html`, which is in the `docs` directory of the software installation, and following the links to the relevant index pages.

Information on available scripts can be found on the Script Center. Information on available software updates can be obtained by choosing Check for Updates from the Maestro menu.

If you have questions that are not answered from any of the above sources, contact Schrödinger using the information below.

E-mail:	help@schrodinger.com
USPS:	Schrödinger, 101 SW Main Street, Suite 1300, Portland, OR 97204
Phone:	(503) 299-1150
Fax:	(503) 299-4532
WWW:	http://www.schrodinger.com
FTP:	ftp://ftp.schrodinger.com

Generally, e-mail correspondence is best because you can send machine output, if necessary. When sending e-mail messages, please include the following information:

- All relevant user input and machine output
- Strike purchaser (company, research institution, or individual)
- Primary Strike user
- Computer platform type
- Operating system with version number
- Strike version number
- mmshare version number

On UNIX you can obtain the machine and system information listed above by entering the following command at a shell prompt:

```
$SCHRODINGER/utilities/postmortem
```

This command generates a file named *username*-*host*-`schrodinger.tar.gz`, which you should send to help@schrodinger.com. If you have a job that failed, enter the following command:

```
$SCHRODINGER/utilities/postmortem jobid
```

where *jobid* is the job ID of the failed job, which you can find in the Monitor panel. This command archives job information as well as the machine and system information, and includes input and output files (but not structure files). If you have sensitive data in the job launch directory, you should move those files to another location first. The archive is named *jobid*-`archive.tar.gz`, and should be sent to help@schrodinger.com instead.

If Maestro fails, an error report that contains the relevant information is written to the current working directory. The report is named `maestro_error.txt`, and should be sent to help@schrodinger.com. A message giving the location of this file is written to the terminal window.

More information on the `postmortem` command can be found in Appendix A of the *Job Control Guide*.

On Windows, machine and system information is stored on your desktop in the file `schrodinger_machid.txt`. If you have installed software versions for more than one release, there will be multiple copies of this file, named `schrodinger_machid-`*N*`.txt`, where *N* is a number. In this case you should check that you send the correct version of the file (which will usually be the latest version).

If Maestro fails to start, send email to help@schrodinger.com describing the circumstances, and attach the file `maestro_error.txt`. If Maestro fails after startup, attach this file and the file `maestro.EXE.dmp`. These files can be found in the following directory:

```
%USERPROFILE%\Local Settings\Application Data\Schrodinger\appcrash
```

Index

N

O

P

Q

R

S

T

U

V

W

120 West 45th Street, 29th Floor
New York, NY 10036

101 SW Main Street, Suite 1300
Portland, OR 97204

8910 University Center Lane, Suite 270
San Diego, CA 92122

Zeppelinstraße 13
81669 München, Germany

Dynamostraße 13
68165 Mannheim, Germany

Quatro House, Frimley Road
Camberley GU16 7ER, United Kingdom

SCHRÖDINGER®